Multiple Nonlinear Prediction of S&P500 Returns Using an ANFIS

Master Thesis submitted to

Prof. Dr. Ostap Okhrin

Humboldt-Universität zu Berlin

Ladislaus von Bortkiewicz Chair of Statistics

by

David Winkel

(553765)

February 24, 2015

Abstract

This thesis presents with the ANFIS a concept in machine learning to predict the returns of the S&P500 nonlinearly. Following Welch and Goyal (2008) the benchmark for the performance of the return predictions is the returns' historical average. The ANFIS is applied to data captured over 1-year and 2-year periods. The ANFIS fails to outperform the historical average using 1-year data. The ANFIS using 2-year data however is able to outperform the historical average.

Keywords:
Fuzzy logic, fuzzy inference systems, neural networks, ANFIS, machine learning, return prediction

Zusammenfassung

Diese Arbeit präsentiert mit dem ANFIS ein Konzept aus dem Machine Learning mit dessen Hilfe die Rendite des S&P500 nichtlinear vorhergesagt wird. In Anlehnung an Welch and Goyal (2008) wird als Vergleichsgröße zur Renditevorhersage der historische Durchschnitt der Rendite verwendet. Das ANFIS wird auf Daten angewendet, welche über 1-jährige Zeiträume und 2-jährige Zeiträume erhoben wurden. Bei der Verwendung der Daten der 1-jährigen Zeiträume gelingt es mit dem ANFIS nicht den historischen Durchschnitt der Rendite als Vergleichsgröße zu schlagen. Angewendet auf die Daten der 2-jährigen Zeiträume ist es jedoch möglich die Vergleichsgröße zu schlagen.

Schlüsselwörter:
Fuzzylogik, Fuzzy Inferenz System, Neurales Netzwerk, ANFIS, Machine Learning, Renditevorhersage

Acknowledgements

First of all I would like to thank Prof. Dr. Ostap Okhrin who gave me the opportunity in writing this thesis. Further thanks go to Prof. Dr. Wolfgang Härdle for his guidance over the whole course of my master's program.

Special thanks goes to my mother who supported me during the experience of writing this thesis as only a mother can do. I also have to thank my sister and my father who provided me both with all the assistance anyone can hope for.
And last but not least I want to thank Ona for her support and all the creative distraction she provided.

Contents

List of Figures

List of Tables

Nomenclature

ACF	autocorrelation function
ANFIS	adaptive neuro-fuzzy inference system
ANN	artifical neural network
AR	autoregressive
CPI	consumer price index
FIS	fuzzy inference system
HLR	hybrid learning rule
LSE	least square estimation
MCP	McCulloch-Pitts
MF	membership function
MLP	multilayer perceptron
MSE	mean squared error
P/E	price-earning
SSR	sum of squared residuals

1 Introduction

1.1 Motivation

What moves the stock markets? This question is as old as the stock markets themselves.

For a long time the academic view on this question was coined by the random walk hypothesis. Originally examined by Kendall and Hill (1953) and further developed by Fama (1965) this theory states that stock prices move randomly. Thus it is not possible to predict the movements in any way. Another influential theory, also consistent with the random walk hypothesis, was the efficient markets theory by Fama (1970). Based on the efficient markets theory many authors denied return predictability since it would imply market inefficiency.

In contrast many successful practitioners like value-oriented investors as Graham and Dodd (1934) stated that certain variables like fundamental ratios can predict stock returns over long time horizons.

In the late 1980s however the academic paradigm of unpredictable returns was challenged by several papers showing statistical evidence for the predictability of returns. Fama and French (1988a) as well as Campbell and Shiller (1988) found that dividend yields are positively correlated with subsequent returns. Their studies concluded a predictability especially over long time horizons.

Also correlations between subsequent stock returns and other variables have been found such as short-term and long-term US treasury yields by Campbell (1987).

The research continued in the 1990s with studies finding other significant explanatory variables such as the book-to-market ratio by Pontiff and Schall (1998) and Kothari and Shanken (1997) and also the price-earning (P/E) ratio by Lamont (1998). Due to the large number of studies stating return predictability the prevailing tone in the academic literature at the end of the 1990s is best summarized by Cochrane (1999) calling the predictability a "new fact in finance".

Recent studies in the 2000s however began to cast doubt on the studies finding return predictability. Goyal and Welch (2003) for example examined the dividend yield as a explanatory variable and found a poor out-of-sample performance of the model. They argued that the predictability can only be found in pre-1990 data. In a further study Welch and Goyal (2008) re-examined the empirical evidence of various studies using variables such as the P/E ratio, the book-to-market ratio or long-term US treasury yields to predict stock returns. Again they found predictability only in certain time periods but a poor out-of-sample performance. Other authors like Butler, Grullon, and Weston (2005) and Campbell and Thompson (2008) also confirmed the often poor out-of-sample performance of linear regression models. The linear regression framework

mostly used was also point of criticism. So Chen and Hong (2010) and Campbell and Shiller (1998) emphasize that the true relation between valuation ratios and long-horizon returns might be nonlinear.

This thesis examines models addressing some points of criticism found in the recent studies. The examined models are used to predict the returns of the S&P500. This thesis challenges Welch and Goyal (2008) who stated the superiority of the historical average as prediction over regression models. The objective is to find a regression model able to outperform the historical average as a predictor for returns. Additionally a good out-of-sample performance of the found model shall not only be limited to a certain time period but be valid for any time period.

This thesis examines different models of the so called adaptive neuro-fuzzy inference system (ANFIS). The ANFIS was proposed by Jang (1993) and is a concept in machine learning based on an artificial neural network (ANN) capable of modelling nonlinear relationships. It utilizes the principles of a fuzzy inference system (FIS). A strength of the ANFIS is its suitability for the hybrid learning rule (HLR) which has computational advantages over other methods for parameter identification.

At the end of this thesis the question is raised whether the ANFIS and ANNs in general are suited for financial applications.

The presented thesis is structured into four chapters. The current chapter 1 describes the motivation and gives an overview of the thesis. Chapter 2 introduces all concepts necessary to understand the ANFIS. These concepts are fuzzy logic, fuzzy inference systems, the ANN and learning methods for the ANN. Chapter 3 presents the results of the prediction of the S&P500 returns by using the ANFIS. Chapter 4 summarizes the findings of this thesis and discusses its results.

2 Methodology

2.1 Logic

This section is a short introduction to traditional logic which builds the foundation of fuzzy logic.

Logic is the science of reasoning. Reasoning in the context of logic describes the act of inferring. To make inference a so called argument is examined. Arguments are a collection of statements. A statement is a declarative sentence. In the traditional two-valued logic a declarative sentence can only take two truth values, *true* or *false*. An example for a declarative sentence is "God exists". This sentence is capable of being either true or false.

In an argument some of the included statements, so called premises, are used to give reason to accept another statement, the so called conclusion. The premises can be seen as the input of an inference process and the conclusion as the process' output. An example for an argument would be:

premise 1	All men are mortal.	}	Input
premise 2	Socrates was a man.		
conclusion	Socrates was mortal.	}	Output

There are different structures of an argument. These structures are called inference rules. In the following two important inference rules are introduced:

1. One of the most commonly used inference rules in logic is the modus ponens. It consists of two premises, one in the form of "If P then Q" and another in the form of "P", and returns the conclusion "Q". An example for an argument which fits the form of modus ponens is:

premise 1	If it is raining then the street is wet.
premise 2	It is raining.
conclusion	The street is wet.

2. Another commonly used inference rule is the modus tollens. It also consists of two premises, one in the form of "If P then Q" and another in the form of "not Q", and returns the conclusion "not P". An example for an argument fitting the form of modus tollens is:

premise 1	If it is raining then the street is wet.
premise 2	The street is not wet.
conclusion	It is not raining.

In the middle of the 19th century the traditional logic evolved through the work of Boole (1854) into a formalistic discipline. Boole brought the two-valued logic into an algebraic structure. The Boolean algebra is an algebra in which the values of the variables are the truth values *true* or *false*, which are usually denoted as 1 or 0. The main operators in the Boolean algebra are conjunction *and*, denoted $\wedge$, the disjunction *or*, denoted $\vee$, and the negation *not*, denoted $\neg$.

The Boolean algebra became fundamental in the development of digital electronics and is the backbone of all electronics and programming languages nowadays.

Nevertheless of its overwhelming application in modern technology there are some limitations in the use of Boolean algebra and the inherent traditional logic.

A problem in the traditional logic shows up in future contingents. Future contingents are statements about future events. Aristotle formulated the problem as follows: There are two statements about future events "Tomorrow there will be a sea battle" and "Tomorrow there will not be a sea battle". Since only these two possibilities exist one of both statements has already to be true today. This would mean nothing can be done to alter the happening of the event. The generalization of this problem leads to the conclusion that every future event is already determined. This conflicted with Aristotle's idea of the own free will and the idea that humans have the power to determine the course of events in the future. So he stated that the laws of logic do not apply to future events.

To deal with Aristotle's paradox of the sea battle, in the early 20th century, the Polish formal logician Łukasiewicz (1920) proposed a logic with three truth values: *true*, *false* and *as-yet-undetermined*. Later Łukasiewicz and Tarski (1930) generalized this idea even further by formulating a logic on n truth values where $n \geq 2$.

Out of these foundations infinite-valued logics such as fuzzy logic and probabilistic logic arose.

2.2 Fuzzy Logic

Fuzzy logic is an infinite-valued extension of the traditional logic. It is based on the mathematical theory of fuzzy sets, which is a generalization of the classical set theory, introduced in a paper by Zadeh (1965). Zadeh observed that the binary logic of computers is not able to deal with subjective human concepts such as "hot" and "cold". Fuzzy sets enable computers to distinguish between certain degrees of hotness. This idea comes close to the way the human perception works. In fuzzy logic a statement gets a degree of truth in between the states *true* or *false*.

Fuzzy sets can also be used in an inference process and build the foundation of the fuzzy logic.

2.2.1 Fuzzy Sets

A set is based on a two-valued logic and has a crisp boundary. A value either belongs to a set or it does not. For example the set

$$\mathrm{A} = \{x \mid x > 5\} \tag{2.2.1}$$

includes all values x, which are greater than the boundary point 5. Otherwise the value x does not belong to the crisp set A. A set with a crisp boundary is called a crisp set in this thesis.

A fuzzy set is a set without a crisp boundary. It is defined as a set of ordered pairs

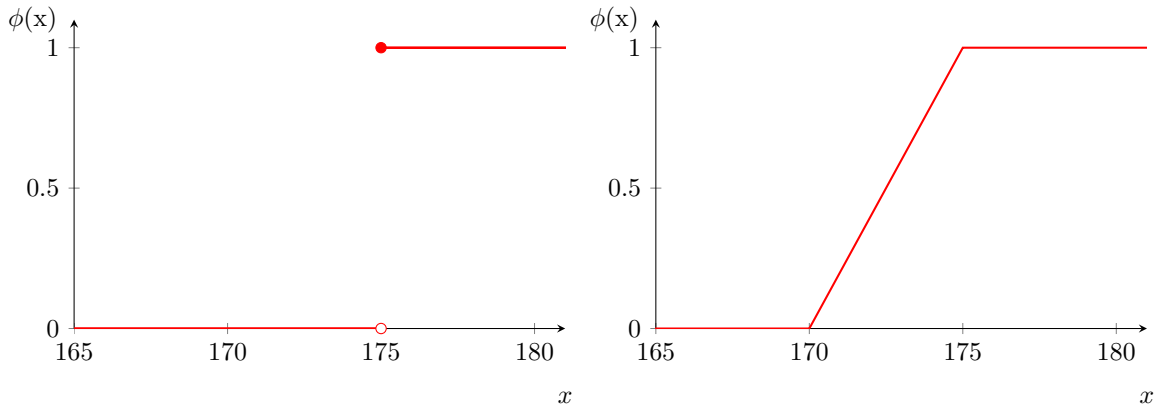
$$\mathrm{B} = \big\{ \big[x, \phi_{\mathrm{B}}(x)\big] \mid x \in \mathrm{X} \big\}. \tag{2.2.2}$$

The function $\phi_{\mathrm{B}}(x)$ is here called the membership function (MF) and defined as

$$\phi_{\mathrm{B}} : \mathrm{X} \to [0, 1]. \tag{2.2.3}$$

It assigns a continuous value between 0 and 1 as a degree of membership $\phi_{\mathrm{B}}(x)$ to each element x in X. The value $\phi_{\mathrm{B}}(x) = 0$ means that x is not a member of the fuzzy set B. A value of $\phi_{\mathrm{B}}(x) = 1$ means that x is a full member of B. Values between 0 and 1 characterize x as a fuzzy member, which means that x belongs to B only partially. A crisp set is a special case of a fuzzy set when $\phi_{\mathrm{B}}(x)$ is equal to an indicator function $\mathbb{1}_{\mathrm{B}}(x)$ which is restricted to values of either 1 or 0.

An example in figure 2.1 illustrates the difference between a crisp set and a fuzzy set. In this example the property "height" of two persons is investigated. At first this property is investigated by using a crisp set with a two-valued logic.



(a) MF of crisp set T_1.

(b) MF of fuzzy set T_2.

Figure 2.1: Comparison of MFs .

A set T_1 is defined to include all persons who are considered to be tall. As a crisp boundary for T_1 a height of 175 cm or greater is chosen. For Peter with a height of 175 cm the statement "Peter is tall" is true because he is a member of set T_1. For Clark

with a height of 174 cm the statement "Clark is tall" is false because he is not a member of set T_1. This assignment seems counterintuitive to the human perception. The same investigation can also be made using a fuzzy set. The statement "Peter is tall" is again true to a degree of 1 and the statement "Clark is tall" is partially true to a degree of 0.8.

This comes closer to the human perception that Peter is tall and Clark is still "somewhat" tall. This example illustrates an advantage of fuzzy logic over the two-valued logic. It is able to come closer to the way human thinking works. Additionally fuzzy logic makes it possible to put natural language in a mathematical framework. The natural language humans use in their every day life consists of linguistic variables.

An example for a linguistic variable is "age" which has different possible realisations. The realisations are called linguistic values. So has "age" linguistic values such as "young", "old" and "very old". Fuzzy sets are a mathematical way to express those linguistic values. The use of fuzzy sets allows to incorporate human knowledge stored in natural language in mathematical models.

The use of linguistic values has also disadvantages though. The individual definition of a linguistic value might differ from person to person. What one person considers as "old" another person might consider as "young". Different people might have different subjective perceptions of the linguistic variable "age". Different definitions of linguistic values lead to different specifications of the fuzzy sets describing these linguistic values. Therefore fuzzy sets and their MFs are highly subjective.

A MF $\phi_{\mathrm{B}}(x)$ of a fuzzy set B can be any function mapping from X to the real interval $[0, 1]$. Nevertheless there are some functions often used as membership functions. Table 2.1 presents some of these parametric functions.

Zadeh (1965) defines in his paper basic operators and relations for fuzzy sets. They are similar to those for crisp sets.

An important relation between fuzzy sets is the containment. Fuzzy set A is contained in fuzzy set B, or alternatively A is called a subset of B, if and only if $\phi_{\mathrm{A}}(x) \leq \phi_{\mathrm{B}}(x)$ for all x, in symbols

$$\mathrm{A} \subseteq \mathrm{B} \Longleftrightarrow \phi_{\mathrm{A}}(x) \leq \phi_{\mathrm{B}}(x). \tag{2.2.4}$$

A union of two fuzzy sets A and B is a fuzzy set C, written as $\mathrm{C} = \mathrm{A} \cup \mathrm{B}$. The MF of C is related to those of A and B by

$$\phi_{\mathrm{C}}(x) = \phi_{\mathrm{A}}(x) \vee \phi_{\mathrm{B}}(x) = \max\big[\phi_{\mathrm{A}}(x), \phi_{\mathrm{B}}(x)\big]. \tag{2.2.5}$$

An intersection of two fuzzy sets A and B is a fuzzy set C, written as $\mathrm{C} = \mathrm{A} \cap \mathrm{B}$. The MF of C is related to those of A and B by

$$\phi_{\mathrm{C}}(x) = \phi_{\mathrm{A}}(x) \wedge \phi_{\mathrm{B}}(x) = \min\big[\phi_{\mathrm{A}}(x), \phi_{\mathrm{B}}(x)\big]. \tag{2.2.6}$$

The complement of a fuzzy set A is a fuzzy set itself, denoted by $\overline{\mathrm{A}}$. The according MF is given by

$$\phi_{\overline{\mathrm{A}}}(x) = 1 - \phi_{\mathrm{A}}(x). \tag{2.2.7}$$

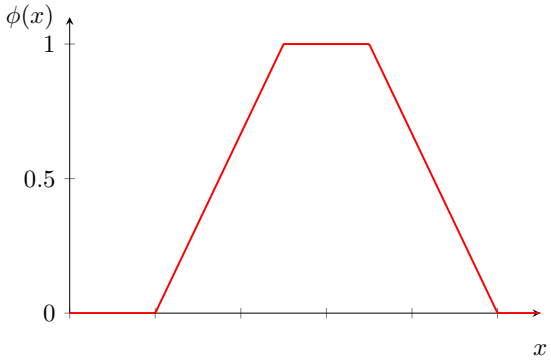
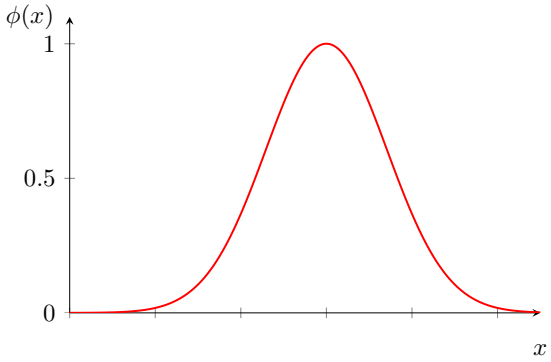
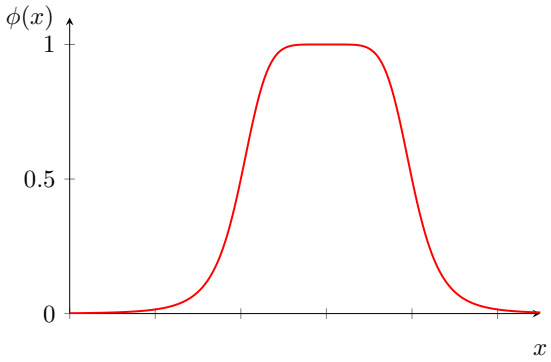
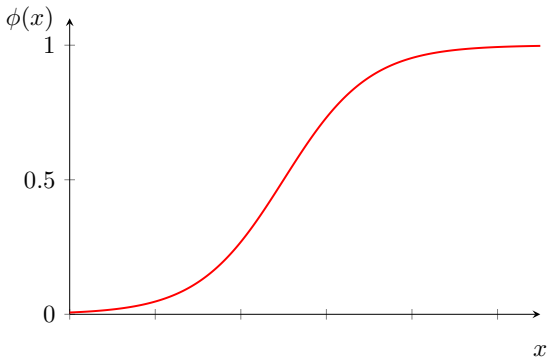
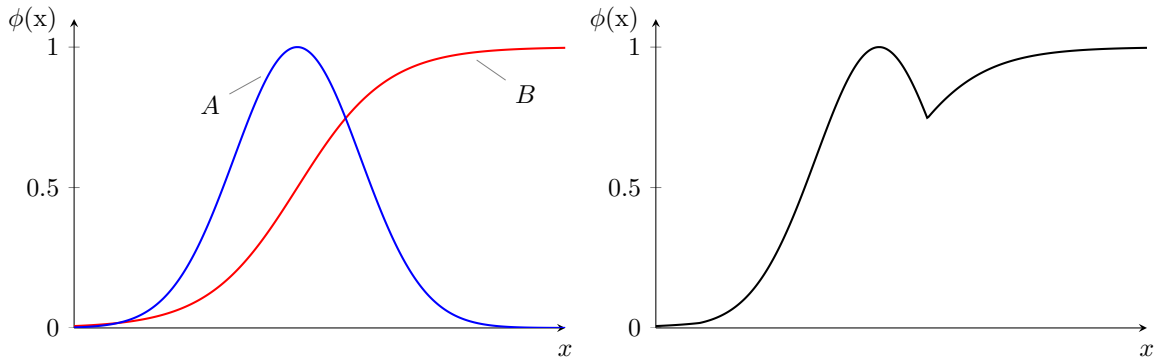
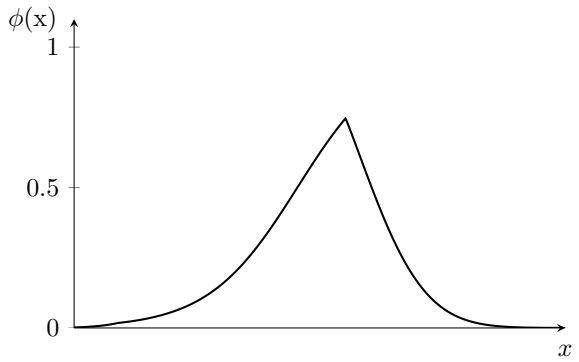
Function	Example
$trapezoid(x; a, b, c, d) = \max\left[\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right]$	
$gaussian(x; c, \sigma) = \exp\left[-\left(\frac{x-c}{\sigma}\right)^2\right]$	
$generalizedbell(x; a, b, c) = \frac{1}{1+\text{abs}\left(\frac{x-c}{a}\right)^{2b}}$	
$sigmoid(x; a, c) = x\left[\frac{1}{1+\exp(-ax+ac)}\right]$	

Table 2.1: Parametric MFs.

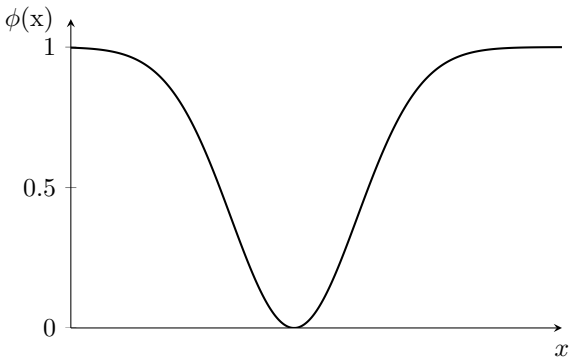


(a) MFs of A and B.

(b) MF of the union of A and B.



(c) MF of the intersection of A and B.



(d) MF of the complement of A.

Figure 2.2: MFs of different operations on fuzzy sets.

The MFs of a union and an intersection of the two fuzzy sets A and B and the MF of the complement of the single fuzzy set A are illustrated in figure 2.2.

Beside the definitions of basic operators and relations just introduced, this thesis also uses different concepts in fuzzy set theory introduced in the following:

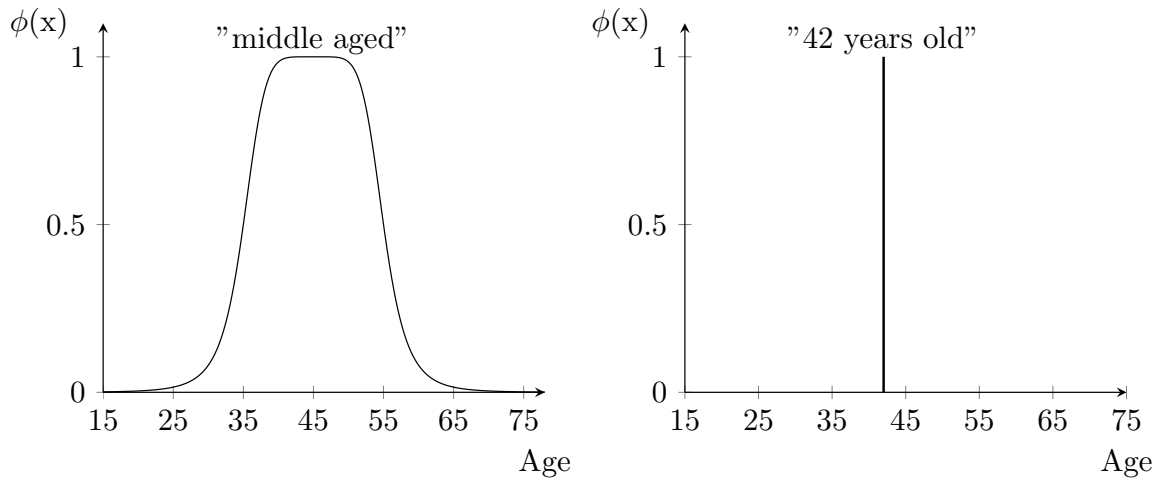
- A singleton is used to represent a crisp value as a fuzzy set A. The singleton A contains only a single point x in X with $\phi_{\mathrm{A}}(x) \neq 0$. For this certain point applies $\phi_{\mathrm{A}}(x) = 1$. Figure 2.3 shows two fuzzy sets in comparison whereby figure 2.3b shows a singleton.

- Fuzzy sets can also be two-dimensional resulting in a MF with two inputs. The two-dimensional fuzzy set A is defined as

$$\mathrm{A} = \big\{ \big[(x, y), \phi_{\mathrm{A}}(x, y)\big] \mid (x, y) \in \mathrm{X} \times \mathrm{Y} \big\}. \tag{2.2.8}$$

  The definition of multidimensional fuzzy sets with more than two dimensions is analog.

- In the application of fuzzy logic it can be necessary to extend the dimension of a fuzzy set. This is done by the so called cylindrical extension. The fuzzy set



(a) MF of the fuzzy set "middle aged". (b) MF of the singleton "42 years old".

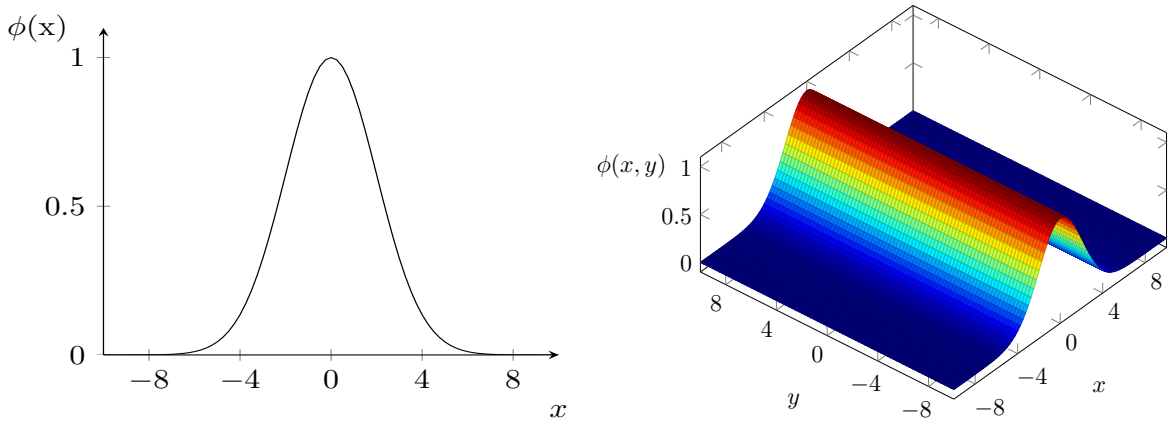
Figure 2.3: Comparison of MFs of two fuzzy sets.

$$c_{\mathrm{Y}}(\mathrm{A})(x,y) = \left\{ \left[(x,y), \phi_{c_{\mathrm{Y}}(\mathrm{A})}(x,y) \right] \mid (x,y) \in \mathrm{X} \times \mathrm{Y} \right\} \tag{2.2.9}$$

is the extension of the fuzzy set A in X, shown in figure 2.4a, to a two-dimensional fuzzy set $c_{\mathrm{Y}}(\mathrm{A})$ in $\mathrm{X} \times \mathrm{Y}$, shown in figure 2.4b. The MFs are related by

$$\phi_{c_{\mathrm{Y}}(\mathrm{A})}(x,y) = \phi_{\mathrm{A}}(x) \quad \forall y \in \mathrm{Y} \tag{2.2.10}$$

meaning the value of $\phi_{c_{\mathrm{Y}}(\mathrm{A})}(x,y)$ is not influenced by y.



(a) MF of fuzzy set A. (b) MF of the cylindrical extension c(A).

Figure 2.4: Cylindrical extension of a fuzzy set.

- The carthesian product of the fuzzy sets A and B, denoted by $\mathrm{A} \times \mathrm{B}$, is a fuzzy set in the dimension $\mathrm{X} \times \mathrm{Y}$

$$\mathrm{A} \times \mathrm{B} = \left\{ \left[(x, y), \phi_{\mathrm{A} \times \mathrm{B}}(x, y) \right] \mid (x, y) \in \mathrm{X} \times \mathrm{Y} \right\} \tag{2.2.11}$$

where the MF is defined as the minimum of the cylindrical extensions of A and B

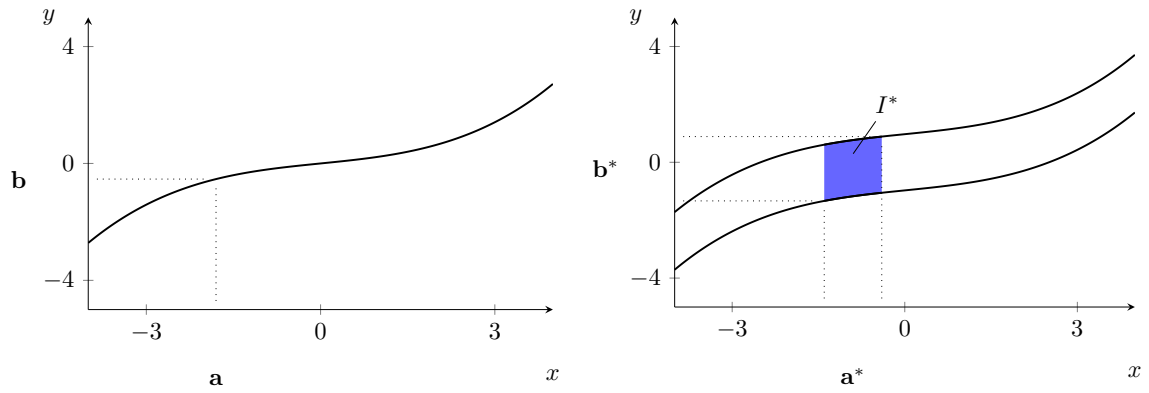
$$\phi_{\mathrm{A} \times \mathrm{B}}(x, y) = \min\left[\phi_{c_{\mathrm{Y}}(\mathrm{A})}(x, y), \phi_{c_{\mathrm{X}}(\mathrm{B})}(x, y) \right] = \min\left[\phi_{\mathrm{A}}(x), \phi_{\mathrm{B}}(y) \right]. \tag{2.2.12}$$

2.2.2 Compositional Rule of Inference

The essential principle behind fuzzy reasoning is the compositional rule of inference. It describes the process of mapping one fuzzy set to another fuzzy set according to a certain relation F. The compositional rule of inference is best explained by generalizing concepts already known.

Supposing a given relation f reflects the relation between X and Y. From the real-valued input a in X can be inferred the real-valued output b in Y by using the relation f, denoted as $f(a) = b$. Figure 2.5a illustrates the relation f and the points a and b.

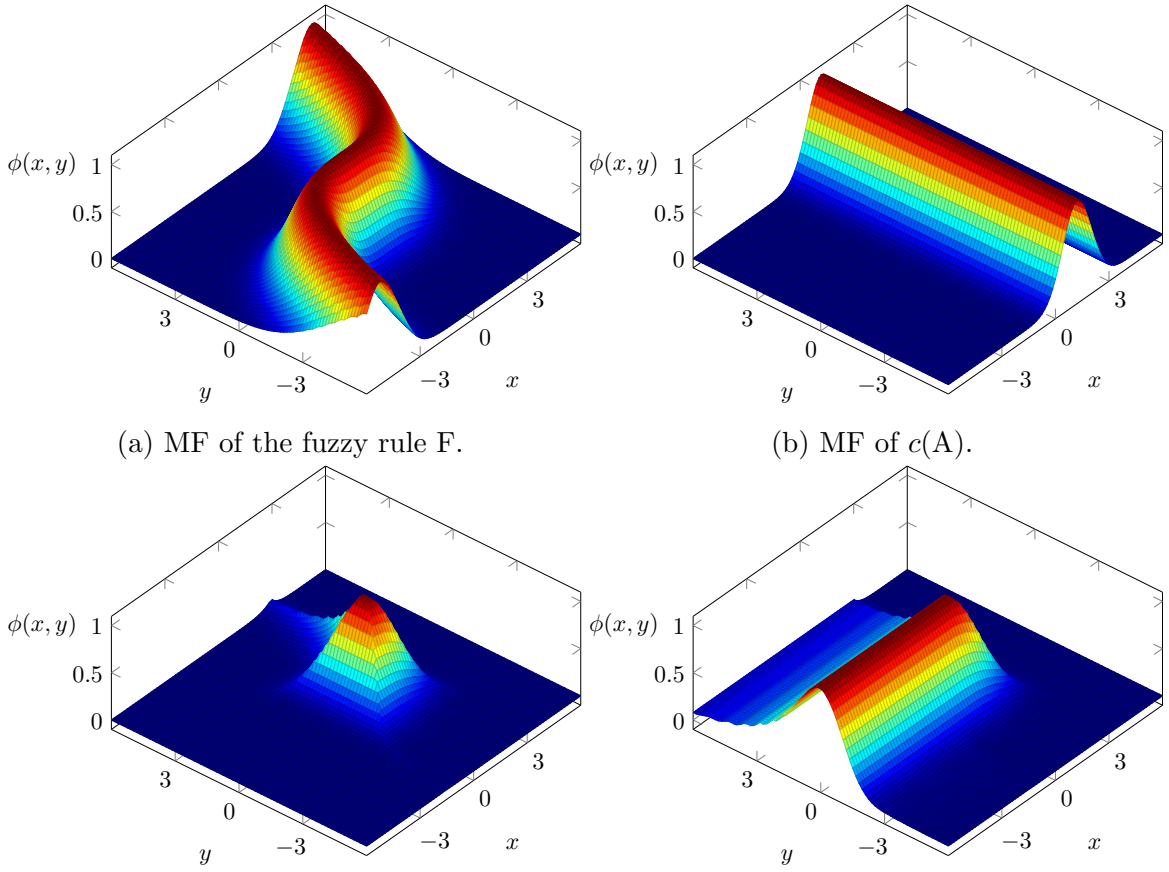
This concept can be extended to the case where the relation f^* is interval-valued, mapping an interval to an interval. Figure 2.5b illustrates the case where interval a^* is mapped to interval b^* by the relation f^*. To find interval b^*, first a cylindrical extension of a^*, denoted $c(a^*)$, is constructed. The cylindrical extension $c(a^*)$ is defined on $X \times Y$, in contrast to a^* which is defined on X. In the second step the intersection I^* of $c(a^*)$ and the interval-valued curve has to be found. In the final step the intersection I^* is projected onto the y-axis yielding the interval b^*.



(a) Relation f. (b) Interval-valued relation f^*.

Figure 2.5: Comparison of two relations.

To generalize even further it is assumed that the relation F maps a fuzzy set to another fuzzy set. Such a fuzzy relation F is also called a fuzzy rule. The fuzzy rule F can be interpreted as a two-dimensional fuzzy set defined as



(a) MF of the fuzzy rule F.

(b) MF of $c(\mathrm{A})$.

(c) MF of I.

(d) Projection of (c) onto Y.

Figure 2.6: Compositional rule of inference.

$$\mathrm{F} = \big\{\big[(x,y), \phi_{\mathrm{F}}(x,y)\big] \mid (x,y) \in \mathrm{X} \times \mathrm{Y}\big\}. \tag{2.2.13}$$

The application of the fuzzy rule F on the input A to receive output B is visualized in figure 2.6. Figure 2.6a shows the MF of a fuzzy rule F on the $\mathrm{X} \times \mathrm{Y}$ space.

For the inference process the fuzzy set A in X is cylindrically extended to the fuzzy set $c(\mathrm{A})$ in the two-dimensional $\mathrm{X} \times \mathrm{Y}$ space. Figure 2.6b shows the MF of $c(\mathrm{A})$ with

$$\phi_{c(\mathrm{A})}(x,y) = \phi_{\mathrm{A}}(x) \quad \forall y \in \mathrm{Y}\,. \tag{2.2.14}$$

Analog to the previous example in figure 2.5b an intersection between the fuzzy rule F and the cylindrical extension $c(\mathrm{A})$ is made. The intersection I is a two-dimensional fuzzy set itself written as

$$\mathrm{I} = \big\{\big[(x,y), \phi_{\mathrm{I}}(x,y)\big] \mid (x,y) \in \mathrm{X} \times \mathrm{Y}\big\}. \tag{2.2.15}$$

The MF can be seen as a function

$$\phi_{\mathrm{I}}(x,y) = g\big[\phi_{c(\mathrm{A})}(x,y), \phi_F(x,y)\big] \tag{2.2.16}$$

of the MF of the two intersected sets. A common choice for $g()$ is the min operator, which leads to

$$\phi_{\mathrm{I}}(x,y) = \min\big[\phi_{c(\mathrm{A})}(x,y), \phi_F(x,y)\big] = \phi_{c(\mathrm{A})}(x,y) \wedge \phi_F(x,y). \tag{2.2.17}$$

The MF of the fuzzy intersection I is shown in figure 2.6c.
The projection of I onto the Y-Axis yields the fuzzy set

$$\mathrm{B} = \big\{\big[y, \phi_{\mathrm{B}}(y)\big] \mid y \in \mathrm{Y}\big\} \tag{2.2.18}$$

visualized in figure 2.6d. Mathematically this can be done by a function $h()$ transforming the function $\phi_{\mathrm{I}}(x,y)$ with a two-dimensional input space back to the function $\phi_{\mathrm{B}}(y)$ with a one-dimensional input space. A common choice for $h()$ is the $\max_x$ operator leading to

$$\begin{aligned} \phi_{\mathrm{B}}(y) = h\big[\phi_{\mathrm{I}}(x,y)\big] &= h\big\{g\big[\phi_{c(\mathrm{A})}(x,y), \phi_F(x,y)\big]\big\} \\ &= \max_x\big\{\min\big[\phi_{c(\mathrm{A})}(x,y), \phi_F(x,y)\big]\big\} \\ &= \bigvee_x\big[\phi_{c(\mathrm{A})}(x,y) \wedge \phi_F(x,y)\big]. \end{aligned} \tag{2.2.19}$$

Due to the choice of the max and min operator this is called the max-min composition and B is represented as

$$\mathrm{B} = \mathrm{A} \circ \mathrm{F} \tag{2.2.20}$$

whereby $\circ$ denotes the composition operator.

2.2.3 Fuzzy If-Then Rules

In the application of fuzzy logic fuzzy if-then rules play a crucial role. Fuzzy if-then rules using linguistic values are widespread in the daily life such as

- If the performance is great then the applause is long.
- If pressure is high then volume is small.
- If the service is good then the tip is high.

A fuzzy if-then rule including the fuzzy set A and B has the general form

$$\text{if } \underbrace{\text{x is A}}_{\text{antecedent}} \text{ then } \underbrace{\text{y is B}}_{\text{consequent}}$$

where the first part of the rule includes the so called antecedent while the second part includes the so called consequent. A fuzzy if-then rule is abbreviated as $\mathrm{R} = \mathrm{A} \rightarrow \mathrm{B}$. As mentioned in the previous section a fuzzy rule can be interpreted as a fuzzy set. In

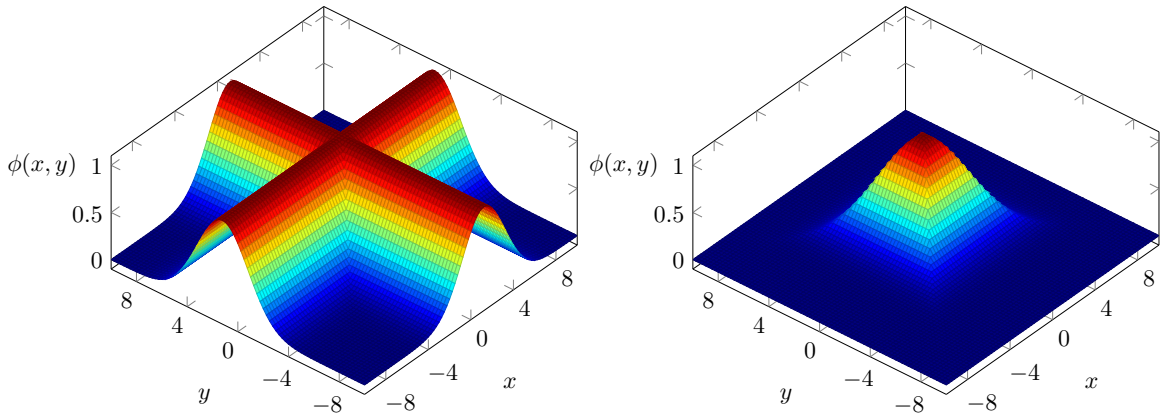
the case of a fuzzy if-then rule including the two fuzzy sets A in X und B in Y, R is defined as

$$\mathrm{R} = \mathrm{A} \rightarrow \mathrm{B} = \big\{ \big[(x, y), \phi_{\mathrm{R}}(x, y)\big] \mid (x, y) \in \mathrm{X} \times \mathrm{Y} \big\}. \tag{2.2.21}$$

Here is $\phi_{\mathrm{R}}(x, y)$ defined as

$$\phi_{\mathrm{R}}(x, y) = f\big[\phi_{\mathrm{A}}(x), \phi_{\mathrm{B}}(y)\big] \tag{2.2.22}$$

where the function f, called the fuzzy implication function, transforms the membership grades of x in A and y in B into membership grades of (x, y) in $R = \mathrm{A} \rightarrow \mathrm{B}$. There are different definitions of the fuzzy implication function used by different authors. Larsen (1980) for example suggests the product operator for the fuzzy implication function. Mamdani and Assilian (1975) by contrast suggest the min operator for the fuzzy implication function. In this thesis Mamdani's and Assilan's definition will be used. The construction of a fuzzy if-then rule R after Mamdani and Assilan is illustrated in figure 2.7. In a first step the fuzzy sets A and B are cylindrically extended as seen in figure 2.7a. In a second step seen in figure 2.7b the fuzzy implication function is applied on the MFs of $c(\mathrm{A})$ and $c(\mathrm{B})$ resulting in $\phi_{\mathrm{R}}(x, y)$.



(a) MFs of cylindrical extension of A and B.

(b) Min operator applied on MFs .

Figure 2.7: Construction of a fuzzy if-then rule.

2.2.4 Fuzzy Reasoning

The following section describes the inference process in fuzzy logic, also called fuzzy reasoning. The concepts of the compositional rule of inference and fuzzy if-then rules already introduced will be used here.

Inference rules in the two-valued logic have different forms. The already introduced modus ponens has the form

Supposing it has to be decided if a banana is ripe. Using the modus ponens and the premises "If the colour is yellow then the ripeness is good" and "The colour is yellow" will lead to the conclusion "The ripeness is good".

premise 1	If x is A then y is B
premise 2	x is A
conclusion	y is B

The human environment however is often hard to classify in a traditional two-valued logical sense. What happens if the banana's colour is not yellow but green-yellow? The human reasoning is able to use the modus ponens in an approximate manner. It would lead from the premises "If the colour is yellow then the ripeness is good" and "The colour is green-yellow" to the conclusion "The ripeness is somewhat good." In a two-valued logical sense this conclusion is not allowed since the statement "The colour is yellow" is false. The reasoning in an approximate manner however is called fuzzy reasoning. Fuzzy reasoning generalizes the inference rules.

The following part introduces fuzzy reasoning using the generalized modus ponens:

- The simplest case of the generalized modus ponens includes a single fuzzy rule with a single antecedent and has the form

premise 1	If x is A then y is B
premise 2	x is A'
conclusion	y is B'

  where the A, A', B and B' are fuzzy sets. The premises "If x is A then y is B" and "x is A'" induce the fuzzy set B' defined as

$$\mathrm{B}' = \mathrm{A}' \circ \mathrm{R} = \mathrm{A}' \circ (\mathrm{A} \to \mathrm{B}) \tag{2.2.23}$$

  or equivalently

$$\mathrm{B}' = \big\{ \big[y, \phi_{\mathrm{B}'}(y)\big] \mid y \in \mathrm{Y} \big\}. \tag{2.2.24}$$

  with the MF , using the max-min composition and equation 2.2.22, of

$$\begin{aligned} \phi_{\mathrm{B}'}(y) &= \max_x \big\{ \min\big[\phi_{\mathrm{A}'}(x), \phi_R(x,y)\big] \big\} \\ &= \max_x \Big(\min\big\{\phi_{\mathrm{A}'}(x), \min\big[\phi_{\mathrm{A}}(x), \phi_{\mathrm{B}}(y)\big]\big\} \Big) \\ &= \max_x \big\{ \min\big[\phi_{\mathrm{A}'}(x), \phi_{\mathrm{A}}(x), \phi_{\mathrm{B}}(y)\big] \big\} \\ &= \bigvee_x \big[\phi_{\mathrm{A}'}(x) \wedge \phi_{\mathrm{A}}(x)\big] \wedge \phi_{\mathrm{B}}(y) \\ &= w_1 \wedge \phi_{\mathrm{B}}(y). \end{aligned} \tag{2.2.25}$$

Figure 2.8 shows the graphical representation of the fuzzy reasoning. Here is w_1 the degree of match between the fuzzy sets A and A′ in the antecedent. The fuzzy rule is then to a degree of w_1 fulfilled. The degree of fulfillment of a rule is also called firing strength. The result of the fuzzy reasoning is the fuzzy set B′ whose MF is represented blue shaded. The MF of B′ is equal to the MF of B clipped at the firing strength.

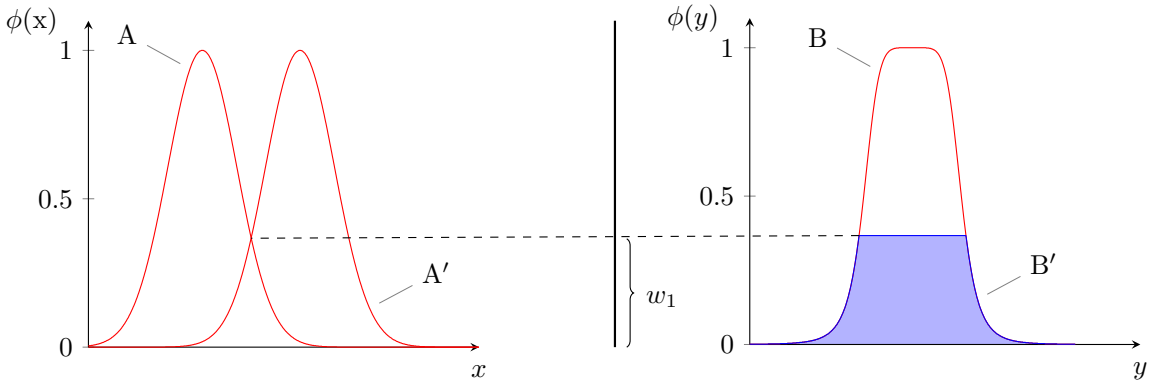


Figure 2.8: Fuzzy reasoning with a single rule and a single antecedent.

- In the case of a single rule with two antecedents the generalized modus ponens is written as

premise 1	If x is A and y is B then z is C
premise 2	x is A′ and y is B′
conclusion	z is C′

where A, A′, B, B′, C and C′ are fuzzy sets. When a fuzzy if-then rule contains multiple antecedents these fuzzy sets are represented by the carthesian product already introduced in section 2.2.1. This leads to

$$\mathrm{C}' = (\mathrm{A}' \times \mathrm{B}') \circ R = (\mathrm{A}' \times \mathrm{B}') \circ (\mathrm{A} \times \mathrm{B} \to \mathrm{C}). \tag{2.2.26}$$

The MF of C′ is defined as

$$\begin{aligned} \phi_{\mathrm{C}'}(z) &= \max_{x,y}\Big\{\min\big[\phi_{\mathrm{A}'\times\mathrm{B}'}(x,y), \phi_{\mathrm{R}}(x,y,z)\big]\Big\} \\ &= \max_{x,y}\Big\{\min\big[\phi_{\mathrm{A}'}(x), \phi_{\mathrm{B}'}(y), \phi_{\mathrm{A}}(x), \phi_{\mathrm{B}}(y), \phi_{\mathrm{C}}(z)\big]\Big\} \\ &= \Big\{\bigvee_x\big[\phi_{\mathrm{A}'}(x) \wedge \phi_{\mathrm{A}}(x)\big]\Big\} \wedge \Big\{\bigvee_y\big[\phi_{\mathrm{B}'}(y) \wedge \phi_{\mathrm{B}}(y)\big]\Big\} \wedge \phi_{\mathrm{C}}(z) \\ &= w_1 \wedge w_2 \wedge \phi_{\mathrm{C}}(z). \end{aligned} \tag{2.2.27}$$

Figure 2.9 illustrates fuzzy reasoning with a single rule and two antecedents. The degree of match between A and A′ is w_1 and the degree of match between B and B′ is w_2. The firing strength of the fuzzy rule is $w_1 \wedge w_2$. The result of the fuzzy reasoning is the fuzzy set B′ that has a MF represented blue shaded.

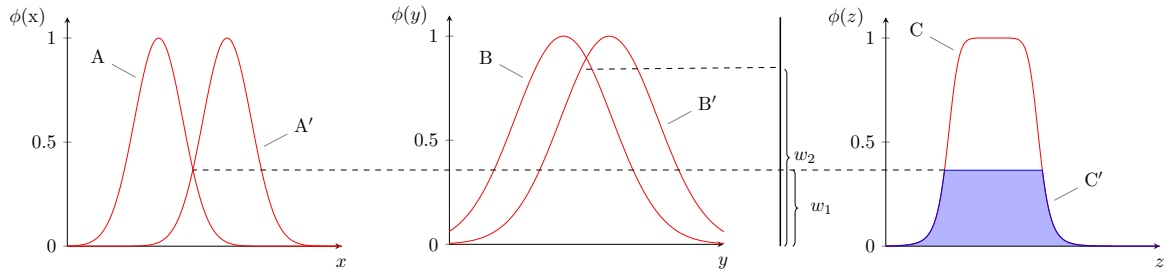


Figure 2.9: Fuzzy reasoning with a single rule and two antecedents.

- A further extension is the case of two rules and two antecedents. The generalized modus ponens has the form

premise 1	If x is A_1 and y is B_1 then z is C_1
premise 2	If x is A_2 and y is B_2 then z is C_2
premise 3	x is A′ and y is B′
conclusion	z is C′

where A_1, A_2, A′, B_1, B_2, B′, C_1, C_2 and C′ are fuzzy sets. Multiple rules R_i with $i = \{1, ..., n\}$ can be treated as the union of the fuzzy rules R_i. Since the max-min composition is distributive over the union operator the result of the fuzzy reasoning is

$$\begin{aligned} C' &= (A' \times B') \circ (R_1 \cup R_2) \\ &= \big[(A' \times B') \circ R_1\big] \cup \big[(A' \times B') \circ R_2\big] \\ &= C'_1 \cup C'_2 \,. \end{aligned} \tag{2.2.28}$$

The MFs of C'_1 and C'_2 can be calculated analog to equation 2.2.27. This results in the MF of C′ written as

$$\phi_{C'}(z) = \max\big[\phi_{C'_1}(z), \phi_{C'_2}(z)\big]. \tag{2.2.29}$$

Figure 2.10 illustrates the fuzzy reasoning with two rules and two antecedents. The MF of the resulting fuzzy set C′ is the maximum of the MFs of the fuzzy sets C'_1 and C'_2.

- Further extensions of the generalized modus ponens with additional antecedents and/or fuzzy rules are analog.

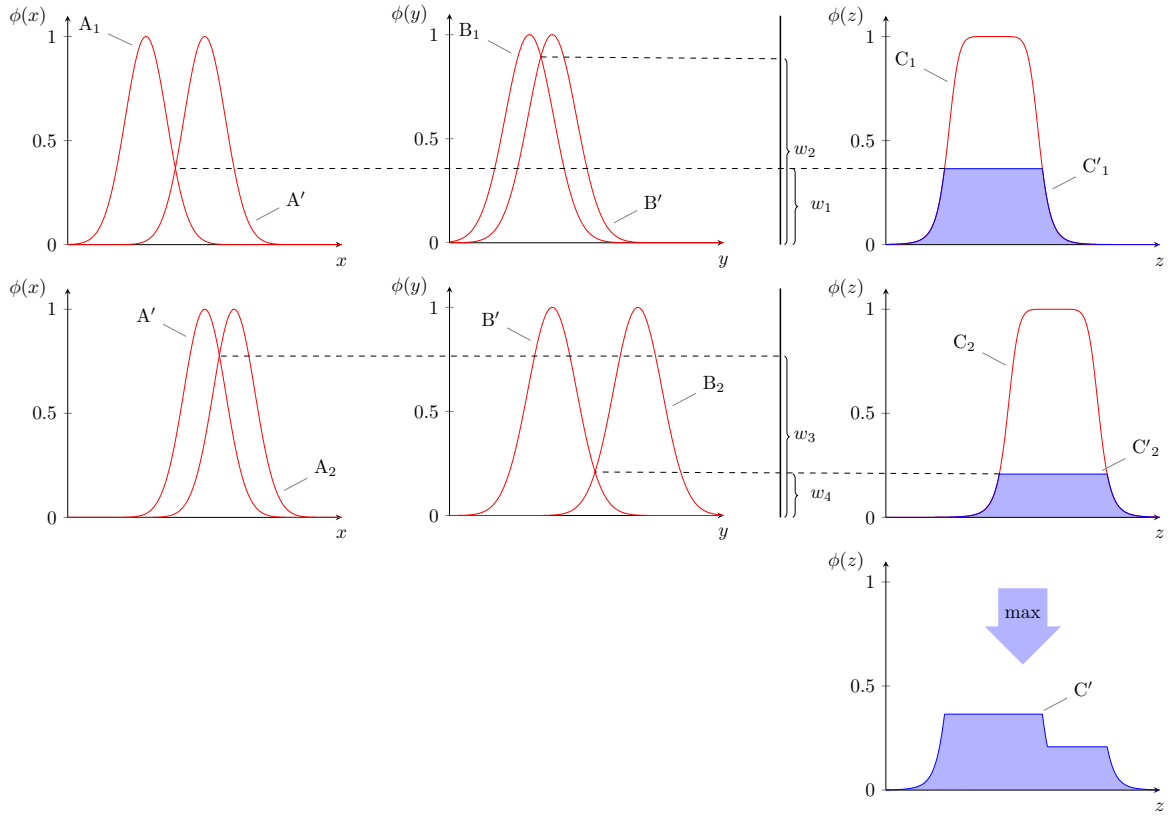


Figure 2.10: Fuzzy reasoning with two rules and two antecedents.

2.2.5 Fuzzy Inference Systems

A fuzzy inference system (FIS) is a computing framework utilizing the already introduced concepts of fuzzy sets, fuzzy if-then rules and fuzzy reasoning. A FIS is an applicable system which performs inference on an input to produce an output.

The following section introduces two commonly used FISs. The first one is the Mamdani inference system. The second FIS to be introduced is the Sugeno inference system.

The Mamdani inference system was originally presented by Mamdani and Assilian (1975) as a solution to control the interaction of a boiler and a steam engine. To construct the inference system Mamdani asked human operators to formulate linguistic if-then control rules which reflected their experience with the boiler/engine system. Using these if-then control rules fuzzy logic can be applied utilizing the operators knowledge.

The inference process in a Mamdani inference system is divided into two steps. The first step is the application of fuzzy reasoning. Figure 2.12 shows an example of a two fuzzy if-then rule Mamdani inference system with two antecedents. Contrary to the example in figure 2.10 this time the input A' in X and B' in Y represent crisp values in form of singletons which have already been introduced in section 2.2.1. The result of the application of the fuzzy reasoning is the fuzzy set C'.

In the second step of the Mamdani inference process the fuzzy set C' is defuzzificated.

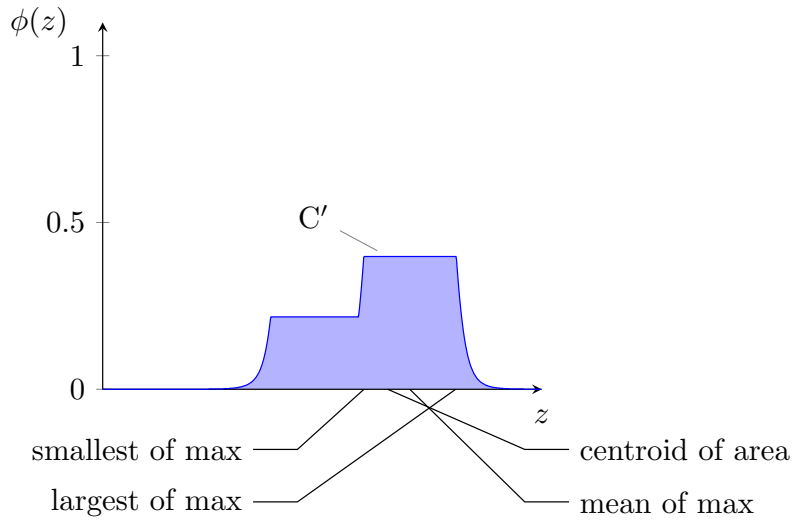


Figure 2.11: Defuzzification methods to obtain a crisp value.

Defuzzification is a method to map a fuzzy set to a crisp value. Figure 2.11 presents some of the existing defuzzification methods. The most common defuzzification method of a fuzzy set C′ is the centroid, which is defined as

$$z_{COA} = \frac{\int_Z \phi_{C'}(z) z \, dz}{\int_Z \phi_{C'}(z) \, dz}. \tag{2.2.30}$$

Thus the Mamdani inference system takes crisp values as input and returns crisp values as output.

The Sugeno inference system was proposed by Takagi and Sugeno (1985). Their idea was to construct a model suited to adapt to a given input-output dataset by modifying the model's parameters. This can be broken down to an optimization problem, which can be solved iteratively. Formerly developed FISs like the Mamdani inference system were not well suited for iterative optimization due to the computationally demanding task of defuzzification in each iteration step. The Sugeno inference system was designed not to depend on defuzzification. It is similar to the Mamdani inference system but the structure of the consequent part in the fuzzy if-then rule, causing the defuzzification, is changed. A typical rule in a Sugeno inference system has the form:

$$\text{If x is A and y is B then } z = f(x, y).$$

where A and B are fuzzy sets in the antecedents and $f(x, y)$ is a crisp function in the consequent. Sugeno and Takagi propose first order polynomials as crisp functions in the consequent part, but the use of other functions is possible.

Figure 2.13 visualizes a two antecedents two fuzzy if-then rule Sugeno inference system. To calculate the output of the system the sum of the weighted consequence func-

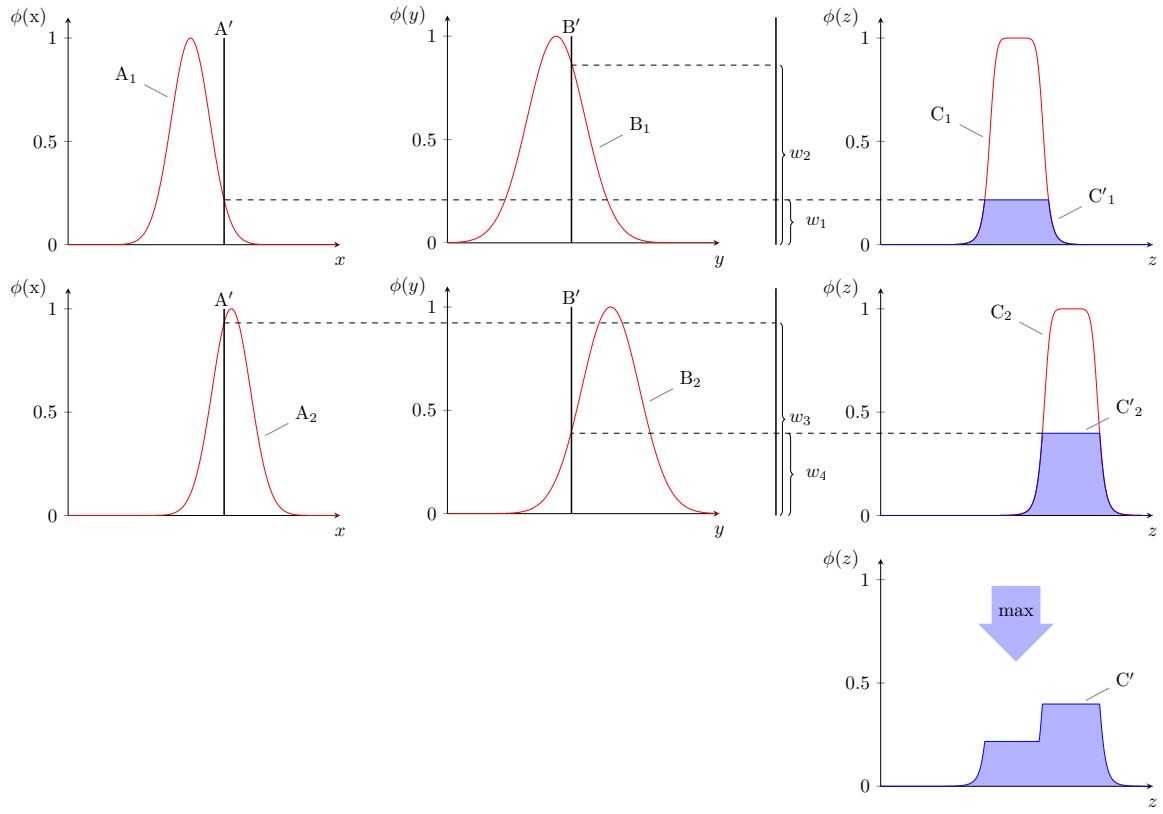


Figure 2.12: Two rule Mamdani fuzzy inference system.

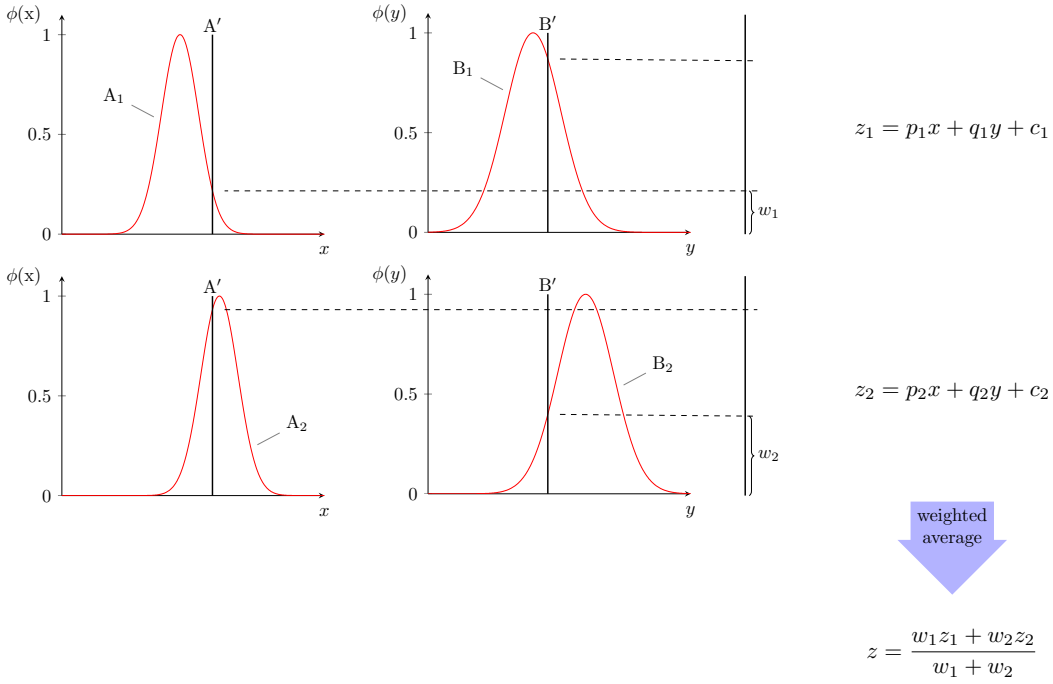


Figure 2.13: Two rule Sugeno fuzzy inference system.

tions is computed. The weights are calculated as the ratio between a rule's firing strength and the sum of firing strengths of all rules.

2.3 Artificial Neural Networks

The following section introduces the artificial neural network (ANN). The first part of this section describes the history of the ANN and some of its properties. The second part will enlighten the connection to fuzzy inference systems leading to the ANFIS.

The development of neural networks has been inspired by the idea to imitate biological nervous systems and replicate how they process information.

The first artificial neuron was proposed by McCulloch and Pitts (1943). In their paper the neuroscientist McCulloch and the logician Pitts tried to replicate how the human brain works. It can produce highly complex patterns by using many interconnected cells. These cells can send signals only in a binary mode, either fire a signal or not fire a signal. The basic idea of their model of a neuron is that $n \in \mathbb{N}$ binary input variables are processed in the neuron. If the sum of these inputs is greater or equal to a certain threshold θ the neuron gives an output of 1. If the sum of the inputs is less than the threshold θ the output of the neuron is 0. Figure 2.14 shows the conceptual structure of a McCulloch-Pitts (MCP) neuron.

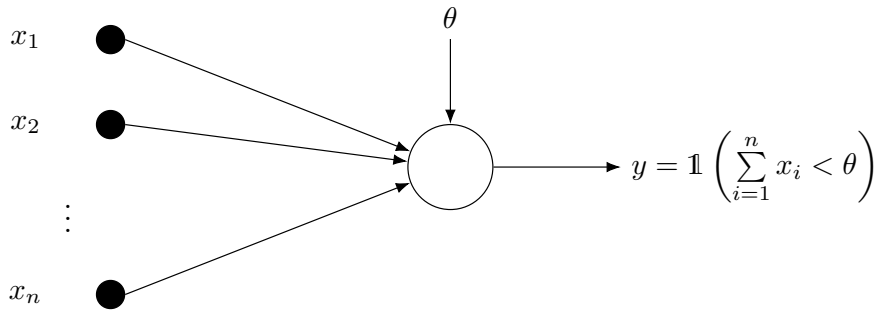


Figure 2.14: Conceptual structure of a MCP neuron.

McCulloch and Pitts showed in their paper the possibility to encode any logical function

$$f : \{0,1\}^n \to \{0,1\} \quad \text{with } n \in \mathbb{N} \tag{2.3.1}$$

by a network of appropriately connected MCP neurons. This means every operation computable by Boolean algebra is also computable by a network of MCP neurons.

An example in table 2.2 shows the truth table for the basic logical OR-function, which can be encoded using a single two-input MCP neuron.

A downside of a network of MCP neurons is that it has to be completely specified before it can be used. Therefore the system's input-output behaviour is completely determined and is fixed after its specification. By contrast biological systems have a flexible input-output behaviour due to their learning ability.

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Table 2.2: Input and output for a two-input MCP neuron with $\theta = 1$, representing the logical OR-Function.

To overcome the limitation of the fixed input-output behaviour the psychologist Rosenblatt (1958) proposed another attempt to model biological neurons. He called his model perceptron. A perceptron also uses a threshold θ and gives binary output. The major difference to the MCP neuron is that the inputs are weighted and that these weights can be modified. By modifying the weights of inputs the perceptron changes its input-output behaviour. The modification of weights is the crucial point that allows learning and enables the perceptron to recognize patterns. A single perceptron is capable of learning and can be trained for example as a classifier for two different groups. Figure 2.15 shows the conceptual structure of a single perceptron.

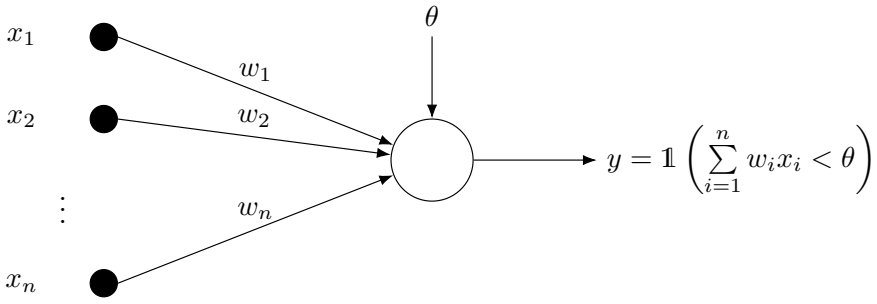


Figure 2.15: Conceptual structure of a perceptron.

The inputs x_i are weighted by w_i with $i \in \{1, ..., n\}$ and then summed up. The sum is compared to a threshold value θ. If the sum is greater than the threshold value the perceptron gives an output of 1. If the sum is smaller than the threshold value the perceptron gives an output of 0. Thus a perceptron is a function with $n + 1$ parameters which maps a n-dimensional input into a binary output

$$g : \mathbb{R}^n \to \{0, 1\} \quad \text{with } n \in \mathbb{N}. \tag{2.3.2}$$

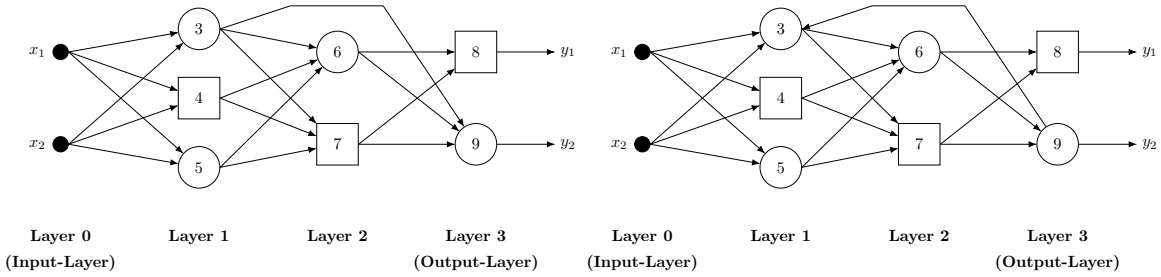
To achieve learning Rosenberg randomly modified the weights by a trial and error principle.

Although initially promising the computational power of the perceptron was questioned in a paper by Minsky and Papert (1969). They showed the inability of a single perceptron to represent a simple nonlinear function such as the XOR-function. They noted that a multilayer perceptron (MLP) – a connected network of perceptrons – would

be able to do so, but that there is no known method to train a MLP. This paper's finding caused a significant decline in interest and funding of neural network research for over a decade resulting in many researchers leaving this field.

Werbos (1974) found a solution to the problem of training a MLP with the backpropagation method. Nevertheless it was not until the mid of 1980s that the neural network research gained popularity again through a further paper about the backpropagation method by Rumelhart, Hinton, and Williams (1986).

This development paired with the progress in computing technology led to the development of the ANN. An ANN is a further generalized idea of a MLP. The ANN consists of multiple layers of so called nodes. Each node represents a node function. In contrast to the perceptrons in the MLP the nodes in an ANN can represent any parameterized function. The input-output behaviour of the entire ANN is determined by the connections of the nodes and the parameters in each node. The ANN can be trained by modifying the parameters in the nodes. The nodes that contain modifiable parameters are called adaptive nodes. The nodes that do not contain modifiable parameters are called fixed nodes. Further graphical representations in this thesis will use squared nodes to represent adaptive nodes and circled nodes to represent fixed nodes.



(a) Feedforward neural network. (b) Recurrent neural network.

Figure 2.16: Comparison of ANNs.

ANNs can be classified into two different groups depending on the directions of their connections. The ANN shown in figure 2.16a is a feedforward neural network. The connections of each node are exclusively directed to higher layers. By contrast figure 2.16b shows a recurrent neural network where a feedback connection between the nodes exists forming a circular path.

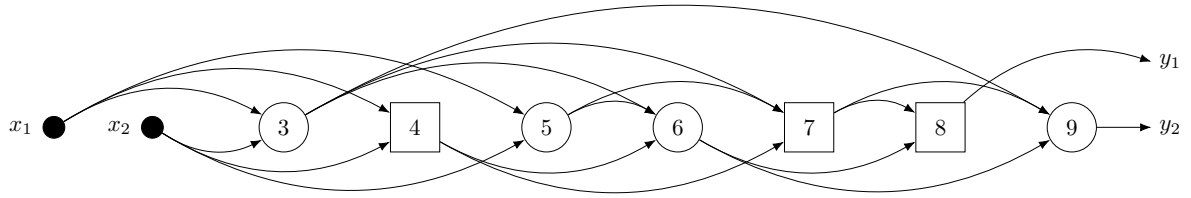


Figure 2.17: Feedforward neural network in its topological order representation.

Each feedforward neural network can also be represented in topological order as seen in figure 2.17. In fact the represented feedforward neural network is equivalent to the one in 2.16a. The topological order representation will be helpful in the later section 2.4 about the learning in an ANN.

For further explanations of the ANN a detailed notation is introduced. The layers in an ANN are numbered by l with $l = \{0, .., L\}$. Layer $l = 0$ is here the so called input-layer, while layer $l = L$ is the so called output-layer. The function $N(l)$ gives the amount of nodes in layer l. Each node in an ANN represents a function, the so called node function. The i-th node function with $i = \{1, .., N(l)\}$ in layer l is denoted by $f_{l,i}$. The output of the i-th node in layer l is denoted by $z_{l,i}$. Figure 2.18 exemplary shows an ANN in the notation introduced.

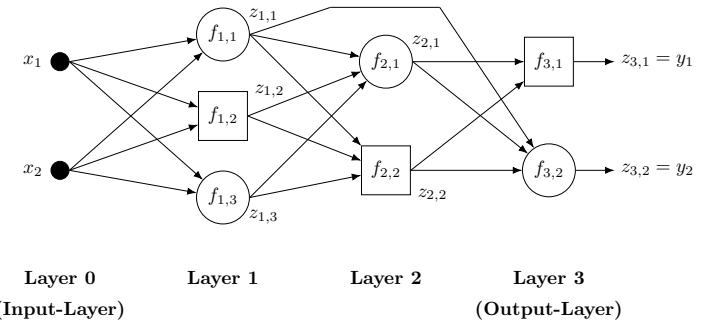


Figure 2.18: Notation of ANN in layered representation.

2.3.1 Adaptive Neuro-Fuzzy Inference System

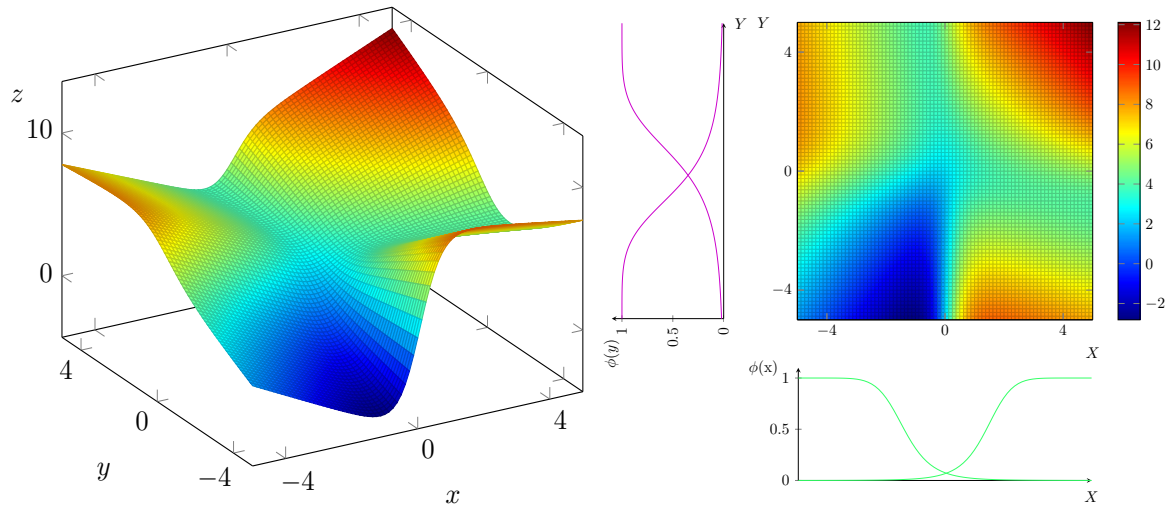
An ANN can also be used as a framework for a FIS. A FIS in ANN representation is called adaptive neuro-fuzzy inference system (ANFIS). By representing a FIS as an ANN the learning methods for ANNs can be applied to identify the parameters in the system.

The following example illustrates how a FIS can be represented as an ANN . The example contains a Sugeno inference system including four fuzzy if-then rules:

- Rule 1: If x is small and y is slow then $z = -x + y + 1$
- Rule 2: If x is small and y is fast then $z = -y + 3$
- Rule 3: If x is large and y is slow then $z = -x + 3$
- Rule 4: If x is large and y is fast then $z = x + y + 2$

Figure 2.19a shows the surface of the Sugeno inference system. Figure 2.19b visualizes the system from a vertical view. The additional graphs on the left and the bottom illustrate the MFs of the fuzzy sets. The purple coloured functions represent the MFs of the fuzzy sets "slow" and "fast" in Y. The green coloured functions represent the

MFs of the fuzzy sets "small" and "large" in X. The MFs divide the shown input space $X \times Y$ roughly into four areas, each mainly described by one of the first order polynomials defined in the consequent of each fuzzy if-then rule. Figure 2.20 illustrates the four rule fuzzy inference system of the example as an ANFIS. The layer 1 represents the MFs of the four fuzzy sets "small", "large", "slow" and "fast". In layer 2 the firing strength w_i with $i = \{1, .., 4\}$ of each of the four rules is calculated by the input of the corresponding MFs. The output of layer 3 will be called normalized firing strength $\bar{w}_i$ and is the ratio of a rule's firing strength to the sum of all rules' firing strength. Layer 4 represents the four polynomials p_i corresponding to the four rules, which are then weighted by the normalized firing strength $\bar{w}_i$ from layer 3. In layer 5 all weighted polynomials are summed up giving the final output of the ANFIS.



(a) Surface of the Sugeno inference system.

(b) MFs and surface from the Sugeno inference system.

Figure 2.19: Sugeno inference system.

This thesis uses an ANFIS based on the Sugeno inference system due to its computational advantage by avoiding defuzzification.

The general structure of a single-output ANFIS based on the Sugeno inference system is described in the following:

- Layer 1 contains the nodes which represent the MFs. These nodes contain the parameters according to the chosen MFs.
- Layer 2 contains the node functions $f_{2,j}$ calculating the firing strength w_j of the j-th rule.
- Layer 3 contains the node functions $f_{3,j}$ calculating the normalized firing strength $\bar{w}_j$ of the j-th rule.
- Layer 4 contains the node functions $f_{4,j}$ representing the consequent function of the j-th rule.

- Layer 5 contains the node function $f_{5,1}$ which sums the weighted consequent functions up and gives the ANFIS output.

A single-output ANFIS based on the Sugeno inference system is then written as

$$\text{anfis}(x_1, ..., x_n) = \sum_{j=1}^{J} \frac{f_{2,j}(x_1, ..., x_n)}{\sum_{i=1}^{I} f_{2,i}(x_1, ..., x_n)} f_{4,j}(x_1, ..., x_n) \tag{2.3.3}$$

where $J = I$ is the amount of rules and $x_1, ..., x_n$ denotes the input variables of the ANFIS.

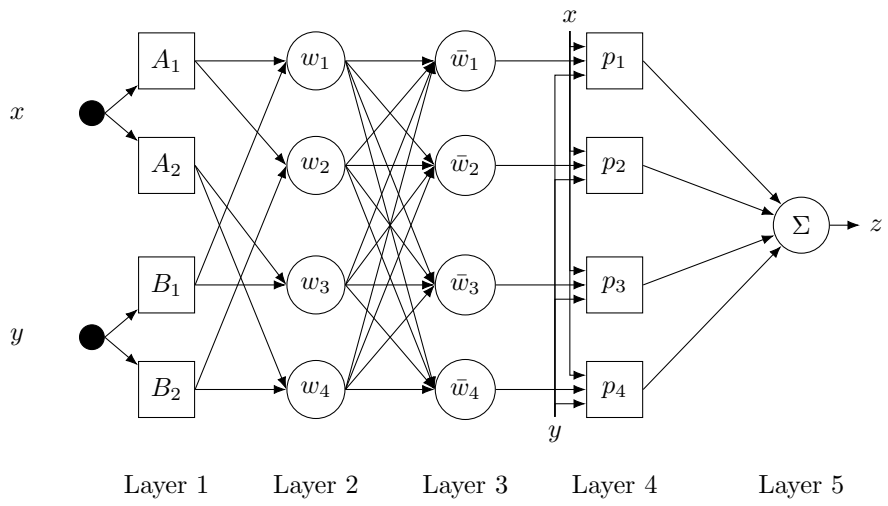


Figure 2.20: ANN representation of the Sugeno inference system: ANFIS.

2.4 Learning

The following section introduces the learning for an ANN. The ANN represents a class of functions F and is supposed to solve a certain task. Learning describes the use of observations to find $f^* \in F$ which solves the task optimally. It is optimal in the sense of minimizing a cost function $C : F \to \mathbb{R}$ such that $C(f^*) \leq C(f)$ for all $f \in F$.

The first part of this section describes the chosen cost function. The second part introduces an optimization algorithm and the third part will present a modification of the optimization algorithm which can be applied to a special case of an ANN, the ANFIS.

2.4.1 Cost Function

An ANN allows to model linear as well as nonlinear relationships between the input and output space. In order to model a certain relationship by the function $f^* \in F$ in an optimal sense the parameter set of f^* has to be found.

How well the ANN reflects the sought-after relationship is measured by the cost function C. The cost function C evaluates the residuals e_p with $p = \{1, .., P\}$, which are defined as the difference between the observed output y_p and the output predicted by the model $\hat{y}_p$:

$$e_p = y_p - \hat{y}_p. \tag{2.4.1}$$

Various different cost functions can be chosen. Following Jang (1993) in this thesis the cost function for the p-th observation, is defined as the sum of squared errors

$$E_p = \sum_{k=1}^{N(L)} e_{p,k}^2 = \sum_{k=1}^{N(L)} (y_{p,k} - \hat{y}_{p,k})^2. \tag{2.4.2}$$

$N(L) > 1$ represents here the case of an ANN with multiple outputs. The term $y_{p,k}$ is the actual observed output for the p-th observation in the k-th variable. The prediction by the ANN for the p-th observation in the k-th output variable is denoted by $\hat{y}_{p,k}$ and equivalent to the k-th output in layer L denoted as $z_{p,L,k}$.

In order to include the cost functions of all P observations the overall cost function is defined as

$$E = \sum_{p=1}^{P} E_p. \tag{2.4.3}$$

2.4.2 Backpropagation Method

Ultimately the ANN's output is determined only by its parameter set. Figure 2.21 shows how a change in a parameter will effect the overall cost function. Therefore the optimization of E is an optimization with respect to the parameters of the ANN.

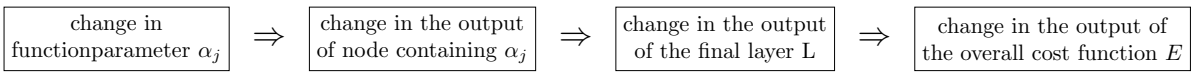
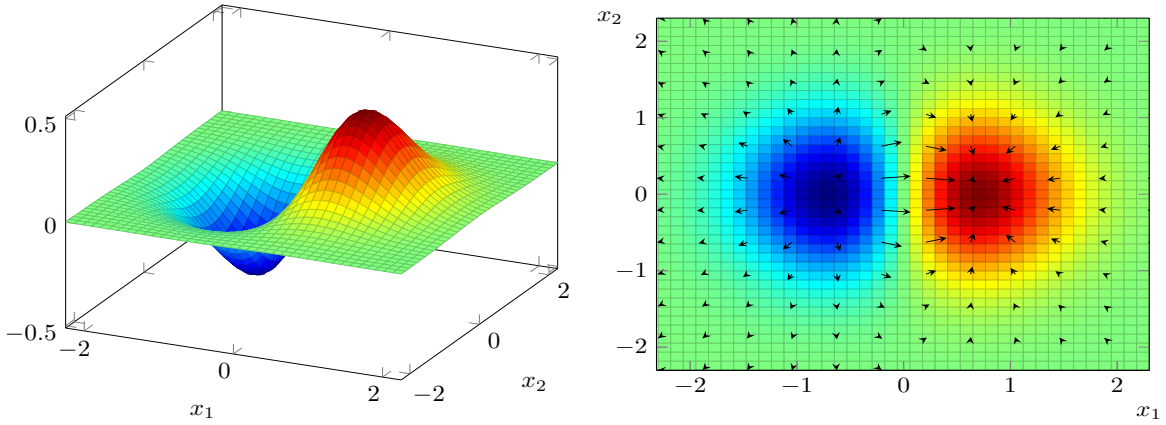


Figure 2.21: Effect of a change in the parameter α_j.

There are different methods for training an ANN. A commonly used method is the backpropagation. The backpropagation method utilizes the gradient descent in order to minimize the cost function.

To understand the backpropagation method first the intuition behind the gradient descent has to be explained. A gradient is the generalization of the one-dimensional concept of a function's derivative. For the n-dimensional vector x the gradient $\nabla f(x)$ of



(a) Surface of $g(x_1, x_2) = x_1 e^{-(x_1^2+x_2^2)}$.

(b) Gradients of $g(x_1, x_2) = x_1 e^{-(x_1^2+x_2^2)}$.

Figure 2.22: Gradient visualisation.

a differentiable, scalar-valued function $f(x)$ is the vector containing all n partial derivatives of the function f. An important property of the gradient is that it points in the direction of the function's greatest rate of increase. The magnitude of the gradient will determine how fast the function is increasing. Figure 2.22 illustrates the gradient. While figure 2.22a shows the examined function $g(x_1, x_2)$, figure 2.22b exemplary illustrates the gradients as directed arrows from a horizontal view. The arrows' length stand for the magnitude of the gradient.

The minimization algorithm gradient descent utilizes the properties of the gradient. It uses the negative gradient which points in the opposite direction of the greatest rate of increase. This is in fact the direction of the greatest rate of decrease in the function. The idea is to create a sequence which "wanders" in each iteration step a step further in the direction of the greatest rate of decrease until finally a minimum is reached. If a minimum is reached the sequence $(x_1, x_2, ...)$ will converge. Formally the sequence is defined as

$$x_{n+1} = x_n - \eta \nabla f(x_n) \quad \text{, with } n \geq 0. \tag{2.4.4}$$

The initial value for x_0 may be a first guess for the coordinates of a minimum of f. The value η is called the learning rate and determines the step size for the negative gradient in each iteration. The elements of the sequence satisfy for small enough η

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq ... \quad . \tag{2.4.5}$$

The following part introduces the backpropagation method, which utilizes the just presented gradient descent. In the first step of the backpropagation method for each node in the ANN an error signal has to be calculated. The error signal is defined as the derivative of the cost function E_p with respect to $z_{l,i}$, which is the output of the i-th node in layer l. Werbos (1974), who introduced the backpropagation method, points out the limitations of using the ordinary partial derivative for networks with dependent

variables. He defines the ordered derivative, "which represents the total change in a later quantity which results when the value of an earlier quantity is changed, in an ordered system."

Ordered derivatives can be calculated by what Werbos called the chain rule for ordered derivatives. For the simple case of a feedforward network with just one node per layer this translates to

$$\frac{\partial^+ z_{out}}{\partial z_i} = \underbrace{\frac{\partial z_{out}}{\partial z_i}}_{\text{direct effect}} + \sum_{j>i} \underbrace{\frac{\partial^+ z_{out}}{\partial z_j}\frac{\partial z_j}{\partial z_i}}_{\text{indirect effect}}. \tag{2.4.6}$$

The example in figure 2.23 illustrates the chain rule for ordered derivatives. The figure shows the topological representation of a simple feedforward neural network and the direct effects in the network in form of its partial derivatives. The node functions are defined as follows

$$\begin{array}{lll} f_{out}(z_0, z_2) & = z_{out} & = 5z_0 + 4z_2 \\ f_2(z_0, z_1) & = z_2 & = 3z_0 + 0.5z_1 \\ f_1(z_0) & = z_1 & = 2z_0 \end{array} \tag{2.4.7}$$

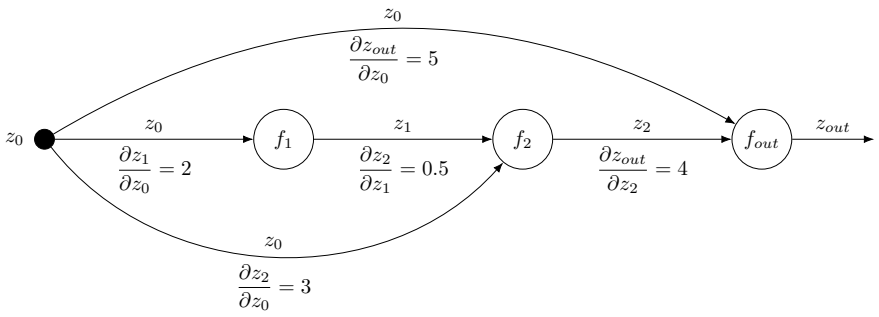


Figure 2.23: Feedforward neural network and its partial derivatives.

The goal in the example is to measure the effect a change in z_0 has on the output of the network z_{out}.

Using equation 2.4.6 the ordered derivatives of z_{out} according to the nodeoutputs z_i are

$$\begin{array}{lll} \frac{\partial^+ z_{out}}{\partial z_2} = \frac{\partial z_{out}}{\partial z_2} & & = 4 \\ \frac{\partial^+ z_{out}}{\partial z_1} = \frac{\partial z_{out}}{\partial z_1} + \frac{\partial^+ z_{out}}{\partial z_2}\frac{\partial z_2}{\partial z_1} & = 0 + 4 \cdot 0.5 & = 2 \\ \frac{\partial^+ z_{out}}{\partial z_0} = \frac{\partial z_{out}}{\partial z_0} + \frac{\partial^+ z_{out}}{\partial z_2}\frac{\partial z_2}{\partial z_1} + \frac{\partial^+ z_{out}}{\partial z_1}\frac{\partial z_1}{\partial z_0} & = 5 + 4 \cdot 3 + 2 \cdot 2 & = 21. \end{array} \tag{2.4.8}$$

By contrast the partial derivative of a function with respect to a variable is derived by holding the other input variables constant which leads to

$$\frac{\partial z_{out}}{\partial z_0} = \frac{\partial f_{out}(z_0, z_2)}{\partial z_0} = 5. \tag{2.4.9}$$

By using only the partial derivative it would be assumed that there are no indirect effects in a feedforward neural network, which is in general not the case. Therefore the ordered derivative has to be used to examine the effect of a change in one variable to another. In order to solve equation 2.4.6 the ordered derivatives of the network have to be solved backwards beginning from the last element of the network as done in the equations 2.4.8.

To calculate the overall cost function in an ANN first the error signals for each node have to be calculated. There are two different cases to distinguish. The first case describes the situation for the error signals of the nodes in the last layer L of the ANN. The last layer's outputs $z_{L,i}$ with $i = \{1, ..., N(L)\}$ can effect E_p due to the structure of the ANN only directly. Calculating the ordered derivative between the cost function and the i-th output of the ANN by using equation 2.4.6 simplifies to

$$\varepsilon_{L,i} = \frac{\partial^+ E_p}{\partial z_{L,i}} = \frac{\partial E_p}{\partial z_{L,i}}. \tag{2.4.10}$$

The second case describes the situation for the outputs of the inner nodes. The inner nodes are all nodes in the layer l with $0 \leq l < L$. An inner node's error signal, which is the node's effect on E_p, is a linear combination of the error signals of the succeeding layer and defined as

$$\varepsilon_{l,i} = \underbrace{\frac{\partial^+ E_p}{\partial z_{l,i}}}_{\text{error signal layer l}} = \sum_{m=1}^{N(l+1)} \underbrace{\frac{\partial^+ E_p}{\partial z_{l+1,m}}}_{\substack{\text{error signal} \\ \text{layer l+1}}} \frac{\partial z_{l+1,m}}{\partial z_{l,i}}. \tag{2.4.11}$$

Thus each error signal $\varepsilon_{l,i}$ in layer l is the sum of the error signals in layer $l+1$ weighted by $\frac{\partial z_{l+1,m}}{\partial z_{l,i}}$.

In order to calculate the error signal for each node in the ANN the error signal equation 2.4.11 has to be solved sequentially backwards from the output layer L to the input layer 0. This method is called backpropagation due to its backwards calculation procedure.

After the use of the backpropagation all nodes' error signals are known. As already presented in figure 2.21 the overall cost function is ultimately determined by the ANN's parameters. Therefore first the effect of changes in a parameter on the p-th error signal has to be determined as

$$\frac{\partial^+ E_p}{\partial \alpha_j} = \frac{\partial^+ E_p}{\partial z_{l,i}} \frac{\partial z_{l,i}}{\partial \alpha_j} = \varepsilon_{l,i} \frac{\partial z_{l,i}}{\partial \alpha_j} \tag{2.4.12}$$

where α_j with $j = \{1, ..., J\}$ is the j-th parameter in the ANN. The parameter a_j is contained in node function $f_{l,i}$ which gives the output $z_{l,i}$. The effect on the overall cost function for a change in α_j is then defined as

$$\frac{\partial^+ E}{\partial \alpha_j} = \sum_{p=1}^{P} \frac{\partial^+ E_p}{\partial \alpha_j}. \tag{2.4.13}$$

To identify the parameter set minimizing the overall cost function the gradient descent is used according to equation 2.4.4. As gradient the $J \times 1$ vector containing the effect of changes in parameters on the overall cost function according to equation 2.4.13 is used.

2.4.3 Hybrid Learning Rule

A weakness of the backpropagation method is its computational intensity caused by the gradient descent. The computational intensity can be reduced by using the so called hybrid learning rule (HLR) as proposed by Jang (1993). The HLR combines the gradient descent and the least square estimation (LSE). However, the hybrid learning rule is only applicable if the ANN is linear in some of its parameters. The linearity in parameters is crucial for LSE. The set of linear parameters S_2 is a subset of the whole parameter set S of the ANN. It applies

$$S = S_1 \cup S_2 \quad \text{with} \quad S_1 \cap S_2 = \emptyset, \tag{2.4.14}$$

where S_1 contains all the parameters of the whole parameter set that are nonlinear.

To estimate the parameters of S_2 by LSE an equation system

$$\mathrm{Y} = \mathrm{X}\,\beta \tag{2.4.15}$$

is build. The $M \times 1$ vector β contains all of the $M = |S_2|$ elements of S_2. X is a $P \times M$ matrix where a row represents the observed input values of the p-th observation with $p = \{1, .., P\}$ in the training dataset. The $P \times 1$ vector Y contains the observed output data of the training dataset. Since P is usually greater than M the system of linear equations is overdetermined meaning there are more equations than unknowns.

To solve an overdetermined system regression analysis can be used. For this purpose a linear regression model is defined. It states for the p-th observation the relationship between the K input variables $x_{p,k}$ and the output y_p as

$$y_p = \beta_0 + \sum_{k=1}^{K} \beta_k x_{p,k} + \varepsilon_p. \tag{2.4.16}$$

The error ε_p is here defined as the deviation between the observed output y_p and the conditional mean $\mathrm{E}(y_p|x_p)$.

Using all P observations from the training dataset in a linear regression model leads to a system of P equations written in matrix form as

$$\mathrm{Y} = \mathrm{X}\,\beta + \varepsilon. \tag{2.4.17}$$

This is equivalent to equation system 2.4.15 with an additional $P \times 1$ error vector ε. The equation system 2.4.17 can be solved using the method of least square estimation. It minimizes the sum of squared residuals (SSR)

$$SSR = e^\top e = (\mathrm{X}\,\beta - \mathrm{Y})^\top(\mathrm{X}\,\beta - \mathrm{Y}) = \mathrm{Y}^\top\mathrm{Y} - \overbrace{\mathrm{Y}^\top\mathrm{X}\,\beta}^{=\beta^\top\mathrm{X}^\top\mathrm{Y}} - \beta^\top\mathrm{X}^\top\mathrm{Y} + \beta^\top\mathrm{X}^\top\mathrm{X}\,\beta$$
$$= \mathrm{Y}^\top\mathrm{Y} - 2\,\beta^\top\mathrm{X}^\top\mathrm{Y} + \beta^\top\mathrm{X}^\top\mathrm{X}\,\beta \tag{2.4.18}$$

with respect to β leading to

$$\frac{\partial SSR}{\partial\,\beta} = -2\,\mathrm{X}^\top\mathrm{Y} + 2\,\mathrm{X}^\top\mathrm{X}\,\beta \stackrel{!}{=} 0. \tag{2.4.19}$$

This is solved by

$$\hat{\beta} = (\mathrm{X}^\top\mathrm{X})^{-1}\,\mathrm{X}^\top\mathrm{Y}\,. \tag{2.4.20}$$

The closed form solution of equation 2.4.20 however is computationally intensive when calculating the inverse of a large $\mathrm{X}^\top\mathrm{X}$ matrix. An alternative approach to compute the least square estimate of β is a recursive method. A widely adopted formula in the literature for example by Aström and Wittenmark (2011) and Ljung (1998) is

$$\begin{aligned} \beta_{i+1} &= \beta_i + \mathrm{G}_{i+1}\,\kappa_{i+1}(\eta_{i+1}^\top + \kappa_{i+1}^\top\,\beta_i) \\ \mathrm{G}_{i+1} &= \mathrm{G}_i - \frac{\mathrm{G}_i\,\kappa_{i+1}\,\kappa_{i+1}^\top\,\mathrm{G}_i}{1 + \kappa_{i+1}^\top\,\mathrm{G}_i\,\kappa_{i+1}} \quad \text{with } i = \{1, .., P-1\}. \end{aligned} \tag{2.4.21}$$

Here is $\kappa_i^\top$ defined as the i-th row vector of X. The i-th element of Y is denoted as $\eta_i^\top$. The initial conditions are $\beta_0 = 0$ and $\mathrm{G}_0 = \gamma I$, where I is the identity matrix of dimension $\mathrm{M} \times \mathrm{M}$ and γ a large positive number. The least square estimate is then β_P.

In the case of multiple output ANNs equation 2.4.21 still applies except that η_i is the i-th row of matrix Y.

After the introduction of the LSE the HLR can be described in the following. The HLR operates iteratively and updates the parameters in S_1 and S_2 in each iteration step. An iteration step is divided into two parts. In the first part, the so called forward pass, the parameters in S_2 are assumed constant and the parameters in S_1 are estimated according to a LSE. In the second part, the so called backward pass, the parameters in S_1 are assumed constant and the parameters in S_2 are identified using the gradient descent. That way the parameter sets S_1 and S_2 are updated in each iteration. Table 2.3 shows the two passes for each iteration step of the HLR.

The reduction of computational intensity of the HLR in comparison to only using the backpropagation method is caused by a reduction of dimension. Although the HLR also includes the gradient descent it is computationally less demanding since the parameter space searched by the gradient descent is smaller, having only a dimension of $M = |S_2|$ instead of $|S|$ as for the backpropagation method.

	Forward Pass	Backward Pass
Parameters in S_1	Fixed	Gradient Descent
Parameters in S_2	Least Square Estimation	Fixed

Table 2.3: The two passes of the hybrid learning rule.

The HLR is well suited for the identification of parameters in an ANFIS based on a Sugeno inference system. This type of ANFIS contains two groups of parameters as already exemplary shown in figure 2.20. The first group includes the parameters describing the fuzzy sets' MFs in the antecedent of the fuzzy if-then rules contained in S_1. The second group contains the functions' parameters of the consequent function in the fuzzy if-then rules. The consequent function of a first order Sugeno inference system is linear in its parameters. Therefore these parameters are contained in S_2.

3 Application

3.1 Data

To investigate the predictability of returns a dataset provided by Prof. Robert J. Shiller on the economic website of Yale University is used. The dataset contains economic data from January 1871 to December 2012. The set consists of data of the S&P500 including stock prices P_t, dividends D_t and earnings E_t. Additionally the set contains economic data as the consumer price index (CPI), the 1-year US treasury yield $i_{1,t}$ and the 10-year US treasury yield $i_{10,t}$.

In a first step of preprocessing, the data is inflation adjusted by using the CPI. In a second step various ratios are calculated from the dataset. The calculated ratios have been picked referring to several studies as already mentioned in section 1.1.

This leads to a preprocessed dataset containing seven variables to be explained in the following.

1. The variable to be predicted is the log return and defined as

$$r_t = \log\left(\frac{P_t + D_t}{P_{t-1}}\right). \tag{3.1.1}$$

2. The remaining six variables are used to predict the log return. The dividend yield will be used as an explanatory variable and is defined as

$$\text{divyield}_t = \frac{D_t}{P_t}. \tag{3.1.2}$$

3. Another ratio often used is the P/E ratio. It can be interpreted as an estimation of how many periods an investment needs to amortize by its own earnings. It is defined as

$$\text{P/E}_t = \frac{P_t}{E_t}. \tag{3.1.3}$$

4. A weakness of the P/E ratio is its volatility caused by the volatility in earnings. An alternative to the P/E ratio is the smoothed P/E ratio defined as

$$\text{P/E}_{h,t} = \frac{P_t}{\frac{1}{h}\sum_{i=t-h+1}^{t} E_i} \tag{3.1.4}$$

Smoothed earnings are less volatile and are more suited to reflect the average long term earning prospective of an investment. In this thesis the earnings are smoothed over a 6-year period.

5. Another explanatory variable is the lagged log return by one period simply defined as

$$r_{\text{lag},t} = r_{t-1}. \tag{3.1.5}$$

6. The 1-year US treasury yield $i_{1,t}$ is taken directly from the Shiller dataset.

7. The 10-year US treasury yield $i_{10,t}$ is taken directly from the Shiller dataset.

In the following different models are presented to analyze the previous explained dataset. Before going into detail a method to evaluate the forecasting performance of these models is presented.

3.2 Evaluation Criterion

In the financial literature the most popular measures for the prediction quality are in-sample approaches like the classical R^2, the adjusted R^2 and testing methods. In prediction however the focus of interest is on how well a model works out-of-sample. That is why Nielsen and Sperlich (2003) introduced the R^2_V measure with the feature to evaluate the out-of-sample predictive power of a model. The R^2_V is defined as

$$R^2_V = 1 - \frac{\sum_t (Y_t - \hat{g}_{-t})^2}{\sum_t (Y_t - \hat{Y}_{-t})^2} \quad \text{with } R^2_V \in (-\infty, 1] \tag{3.2.1}$$

where Y_t is the return in period t and $\hat{g}_{-t}$ is the forecast of the model to evaluate. The model to evaluate is estimated by using all available observations up to period $t-1$. $\hat{Y}_{-t}$ is the historical average with all available observations up to period $t-1$. In recent literature the historical average is often used as a benchmark for models predicting returns. For instance Welch and Goyal (2008) state in their study that most known stock prediction models fail to outperform the historical average out-of-sample.

A positive value of R^2_V indicates the model is able to better predict out-of-sample than the benchmark of the historical average. A negative value of R^2_V shows the inability of the model to predict better out-of-sample than the historical average. The strength of the R^2_V is that it directly reflects the out- or underperformance against the benchmark. This explains the popularity of R^2_V in recent return prediction studies e.g. from Campbell and Thompson (2008).

3.3 Autoregressive Model

To predict returns the first model to evaluate is a simple autoregressive (AR) model of first order. This approach has also been followed by Fama and French (1988b). They focused on the mean reverting property of returns and believed it to cause a negative autocorrelation. The AR(1) model is defined as

$$r_t = c + \gamma_1 r_{t-1} + \varepsilon_t. \tag{3.3.1}$$

The model assumes a linear relationship between returns and 1-period lagged returns. The performance of the AR(1) will be evaluated over two different time horizons.

The AR(1) using 1-year lagged returns is not able to outperform the historical average as seen in table 3.3. The explanation can be found in table 3.1 which shows the estimated parameters of the AR(1). The estimate $\hat{\gamma}_1$ is close to zero and has a p - value of 0.6496. Since the used α-level of 5 % is clearly exceeded, the $H_0 : \gamma_1 = 0$ can not be rejected. Thus it can not be statistically proven that the 1-year lagged return has a linear influence on the return. The autocorrelation function (ACF) in figure 3.1b also shows that there is no further significant autocorrelation in other lagged returns. Figure 3.1a shows that the resulting forecasts are very close to the historical average. The forecast of the AR(1) is almost entirely determined by the estimated constant $\hat{c}$ due to the parameter estimate $\hat{\gamma}_1$ close to zero. The estimated constant $\hat{c}$ is almost identical to the historical average.

The second AR(1) model using 2-year lagged returns is also not able to clearly outperform the historical average as seen in table 3.3. The estimate $\hat{\gamma}_1$ of the 2-year AR(1) model is also not statistically significant. Figure 3.1d also shows no significant autocorrelation in other lagged returns. The forecasted returns in figure 3.1c are close to the historical average again caused by the small parameter estimate $\hat{\gamma}_1$.

Parameter	Estimate	Standard Error	t-Statistic	p-Value
c	0.0612973	0.015722	3.89883	0.000221
γ_1	0.0245322	0.0794362	0.308829	0.758404

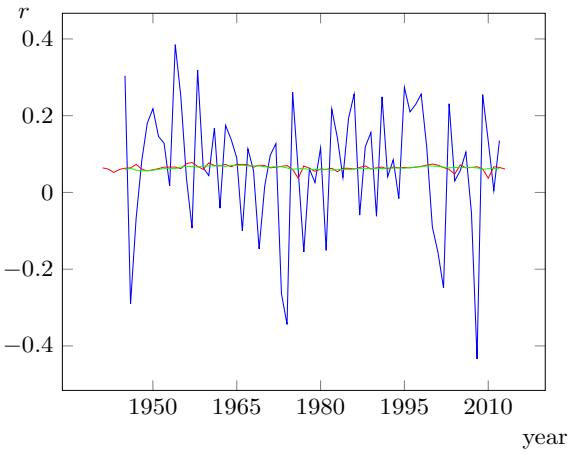
Table 3.1: Estimation result for a 1-year period.

Parameter	Estimate	Standard Error	t-Statistic	p-Value
c	0.12784	0.0392924	3.25356	0.0017
γ_1	-0.0610865	0.132619	-0.460615	0.6469

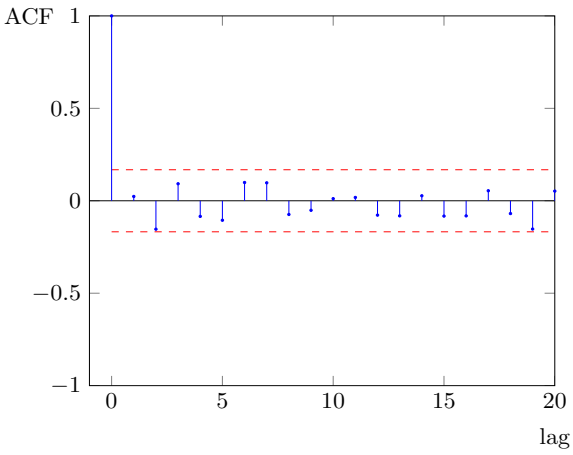
Table 3.2: Estimation result for a 2-year period.

Model	Year Period	R_V^2
AR(1)	1	-0.0113
AR(1)	2	0.0034

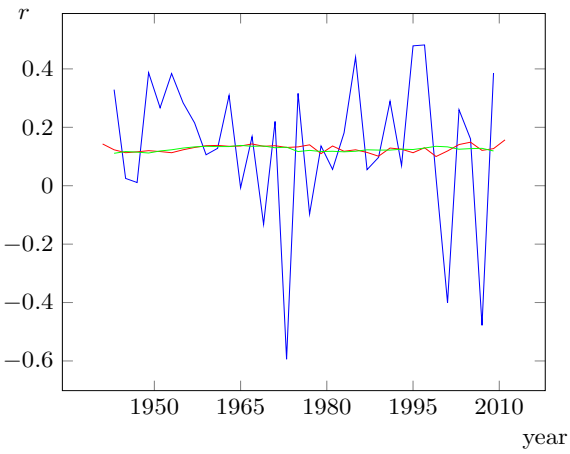
Table 3.3: Results of forecasting by AR(1).



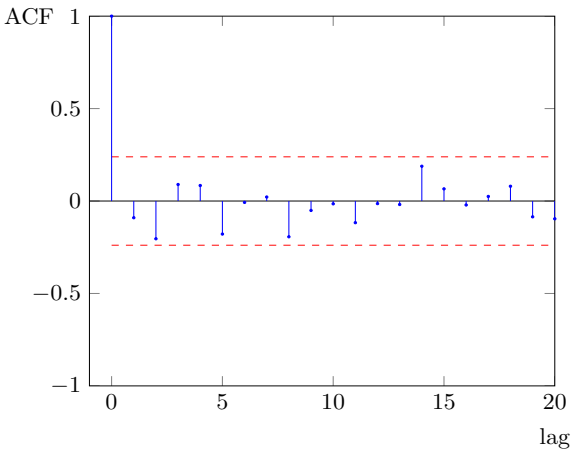
(a) Actual return (blue), 1-year AR(1) forecast (red), historical average (green).



(b) ACF of 1-year lagged returns.



(c) Actual return (blue), 2-year AR(1) forecast (red), historical average (green).



(d) ACF of 2-year lagged returns.

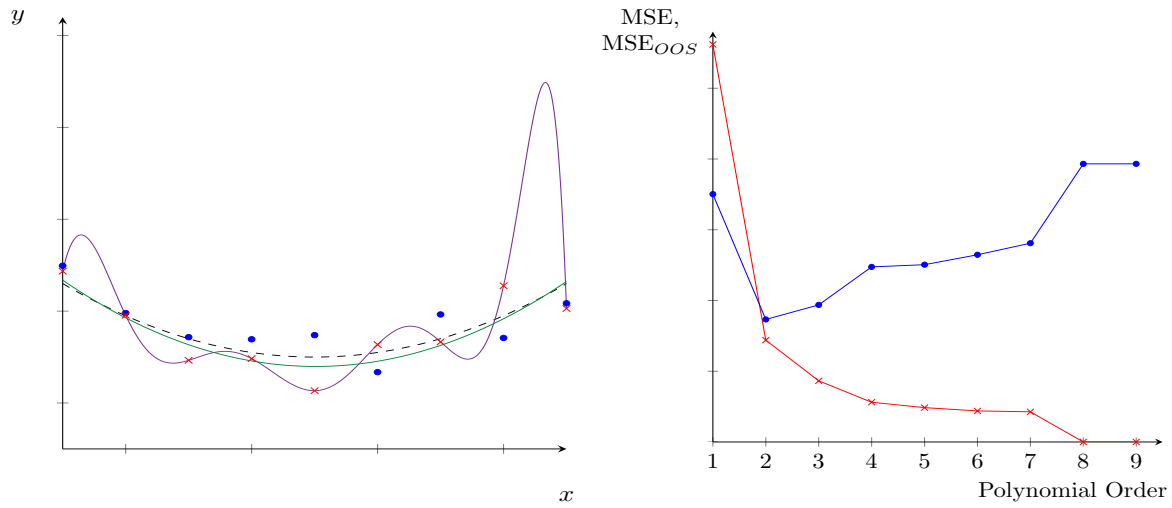
Figure 3.1: Visualisation of the AR(1) results.

3.4 ANFIS Model

3.4.1 General Problems

A simple AR(1) model is not able to outperform the historical average. Therefore the next model studied will be the ANFIS which is more complex and high parametric. Before going into detail of the ANFIS's configuration some general problems with high parametric models have to be addressed.

The higher the amount of parameters in a regression model the less fixed structure is imposed. This allows to model various kinds of nonlinear relationships. An increasing amount of parameters allows the regression model an increasing sensitivity to local observations. Then again this sensitivity makes high parametric models vulnerable to



(a) Fitting observations by polynomials of 2nd order (green) and 9th order (purple).

(b) MSE (red) and MSE$_{OOS}$ (blue).

Figure 3.2: Model overfitting.

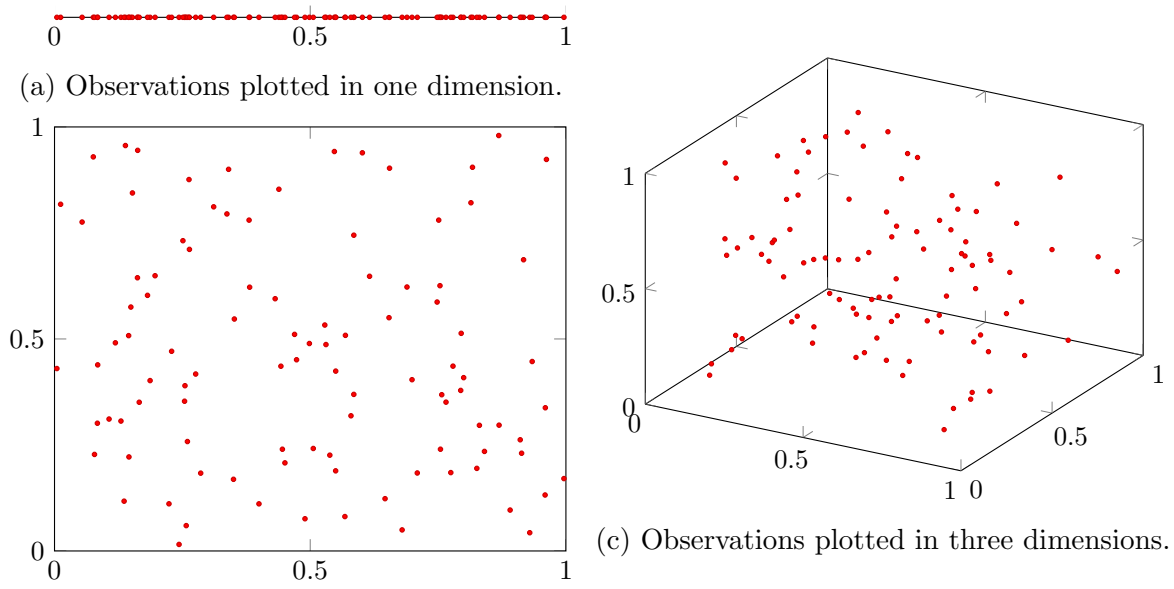
the problem of overfitting. To understand the problem it is important to recall that the objective of a regression model is to estimate the relationship between different variables. Overfitting occurs when the model describes the fluctuation of the random error rather than the relationship itself. Figure 3.2a illustrates the problem. The black dashed line shows the unknown relationship between x and y which shall be estimated by a regression model. The red crosses show 10 observations. From this sample of observations, the training dataset, the relationship has to be estimated. For estimation n-th order polynomial regression models are used and compared. A n-th order polynomial regression model is defined as

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + ... + a_n x_i^n + \varepsilon_i. \tag{3.4.1}$$

The green line in figure 3.2a shows the estimation by a 2nd order polynomial. The purple line shows the estimation by a 9th order polynomial. It can be seen that the 9th order polynomial becomes locally very sensitive to the observations and fits the 10 observations very well. In comparison the 2nd order polynomial reacts rather inflexible to the observations. To evaluate the performance of a model the mean squared error (MSE) can be used. The MSE of a predictor is defined as

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{3.4.2}$$

The MSE for each estimated polynomial is shown in figure 3.2b. The red line shows the MSE between different estimated n-th order polynomials and the red crossed observations. It can be seen that the MSE decreases with increasing order of the polynomial. In spite of the smaller MSE of the 9th order polynomial figure 3.2a illustrates how poorly



(a) Observations plotted in one dimension.

(b) Observations plotted in two dimensions.

(c) Observations plotted in three dimensions.

Figure 3.3: Curse of Dimensionality.

the purple line performs in modelling the black dashed line of the true relationship. By contrast the green line of the 2nd order polynomial comes very close to the black dashed line of the true relationship. The 9th order polynomial is an overfitted model since it describes also the fluctuation of the random error in the observations rather than just the relationship itself. Thus a smaller MSE does not necessarily imply a better statistical model.

In order to avoid overfitting validation techniques can be used. This is done by testing the model's ability to predict data. For this purpose out-of-sample techniques can be used. It is called out-of-sample since these techniques use data which are not in the sample used for the model's estimation. It is assumed that the out-of-sample validation dataset and the training data are generated by the same underlying relationship. A model which estimates the underlying relationship well should be able to predict the observations in the validation dataset well. Figure 3.2a shows the 10 observations of the validation dataset as blue dots. It can be seen that the 9th order estimate performs rather poorly in predicting the blue dots. In contrast the 2nd order polynomial performs better in predicting the blue dots. This can also be seen in figure 3.2b where the out-of-sample MSE, denoted as MSE_{OOS}, between the estimated polynomial models and the validation dataset is shown in the blue line. Here the 2nd order polynomial has the lowest MSE_{OOS}. From the 3rd order polynomial upwards the models begin to overfit and perform worse in predicting.

Since high parametric models are more sensitive to local observations it is important to locally have a sufficient amount of observations to avoid overfitting.

When configuring the ANFIS another problem closely connected to overfitting has

to be taken into account. The Curse of Dimensionality describes the phenomenon that with increasing dimensionality the data becomes sparse due to the fact that the distance between observations increases.

Figure 3.3 illustrates the basic intuition behind the Curse of Dimensionality in a three-dimensional example. In this example 100 observations of three different uniformly distributed variables X_1, X_2, X_3 are known and shown in three different setups. In the first setup shown in figure 3.3a the 100 observations are only examined in the X_1 dimension. It can be seen that the observations are close to each other. In the second setup shown in figure 3.3b the same 100 observations are now additionally examined in the X_2 dimension. Due to the increase in dimension the distance between the observations increases. The third setup in figure 3.3c shows the observations in all three dimensions. The distance between the observations increases further.

The Curse of Dimensionality especially becomes a problem for high parametric models since these models need a sufficient amount of locally close observations for estimating without overfitting.

3.4.2 ANFIS Configuration

The ANFIS is a high parametric model. In order to avoid overfitting the amount of parameters has to be controlled. In this thesis an ANFIS based on a first order Sugeno inference system is used. The amount of parameters depends on several factors here:

1. The amount of input variables: u
2. The amount of MFs per input variable: v
3. The amount of parameters of the chosen MF: w

The total amount of parameters can be calculated as $\mathrm{w} \cdot (\mathrm{v}^{\mathrm{u}}) + (\mathrm{u}+1) \cdot (\mathrm{v}^{\mathrm{u}})$, where v^{u} is the amount of rules in the ANFIS. The first summand contains the set of parameters defining all MFs, the second summand contains the set of parameters defining the first order polynomials in the consequent. The identification of the parameter sets has already been discussed in section 2.4.

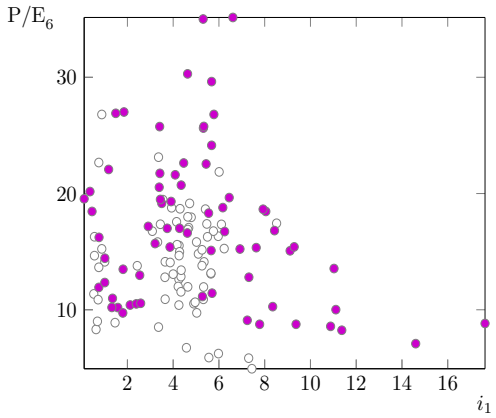
Since the amount of rules grows exponentially to the base of v and the power of u, controlling v and u has a huge impact on the total amount of parameters in the ANFIS model.

Due to the impact of u on the total amount of parameters and to address the Curse of Dimensionality the ANFIS model will only use two input variables. Additionally the amount of MFs v per input variable will also be set to only two. The three-parametric generalized bell-shaped MF be used as chosen MF. Therefore the total amount of parameters for the ANFIS model with the chosen configuration is $3 \cdot 2^2 + (2+1) \cdot 2^2 = 24$.

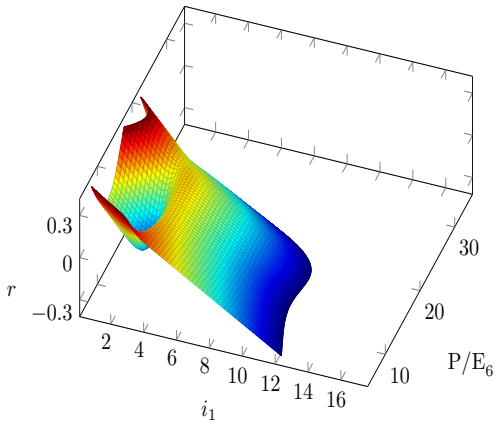
3.4.3 ANFIS Forecasting

The forecasting process is explained exemplary in the next section. In this example the input variables i_1 and $\mathrm{P/E}_6$ are used to forecast the return. The data is plotted in

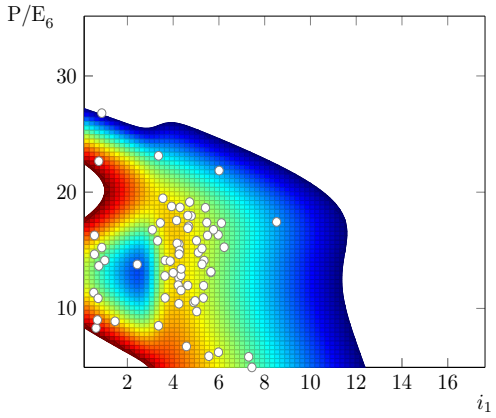
figure 3.4a. It shows 140 observations in form of coloured dots. Although not visualized in this figure each observation also contains information about the return to be forecasted by the model.



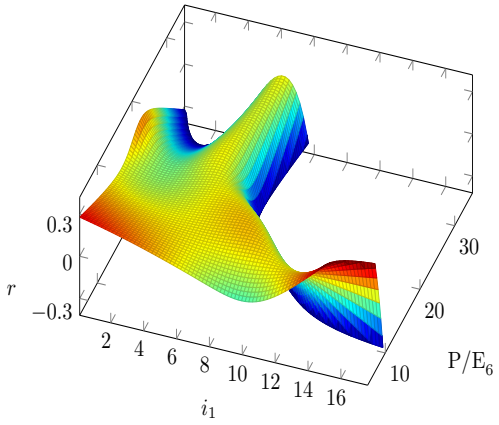
(a) Initial training dataset (white), remaining observations (purple).



(c) With initial 69 observations trained ANFIS.



(b) Initial observations (white) and trained ANFIS



(d) With all available 140 observations trained ANFIS.

Figure 3.4: Visualisation of the forecasting process.

The white dots represent 69 observations from 1871 to 1941 and are the initial training dataset for the ANFIS estimation with input i_1 and $\mathrm{P/E}_6$. The purple dots represent 71 observations from 1942 to 2012 and shall be used to forecast the return out-of-sample.

Figure 3.4b shows the surface of the estimated model used to forecast the return for 1942. The estimation used all 69 observations from the initial training dataset. Figure 3.4c visualizes the used observations and the estimated model from a vertical perspective. For each following period t the model is re-estimated including all observations up to period $t-1$. Thus the size of the training dataset increases by one with each additional period. The prediction for each period t remains out-of-sample though.

As seen in figure 3.4a there are local areas with little or none observations. As

mentioned in section 3.4.1 this causes problems for high parametric models like ANFIS. The estimation in these local areas is based on little or none information contained in the data and tends to give unreasonably high or low estimations. To address this model's weakness the estimation of the ANFIS is capped. Whenever the model gives a higher or lower forecast than the cap values it will be considered as a local area with not enough information for a meaningful ANFIS estimation. The forecast then will be replaced by the out-of-sample historical average. Various cap values have been tested. Cap values of -0.35 respectively 0.35 give the best out-of-sample performance regarding the MSE. Following equation 2.3.3 this leads to a modified ANFIS used in this thesis

$$\text{anfis}_{\text{cap},-t}(x_1, x_2) = \begin{cases} \text{anfis}_{-t}(x_1, x_2) & \text{if } \text{-}0.35 \leq \text{anfis}_{-t}(x_1, x_2) \leq 0.35 \\ \bar{\text{Y}}_{-t} & \text{else} \end{cases} \tag{3.4.3}$$

where anfis_{-t} represents the ANFIS model using for parameter identification all observations up to period $t-1$. $\bar{\text{Y}}_{-t}$ is the return's historical average using all observations up to period $t-1$.

Figure 3.4d shows the last estimated model for predicting the return in 2013 utilizing all 140 observations for estimation.

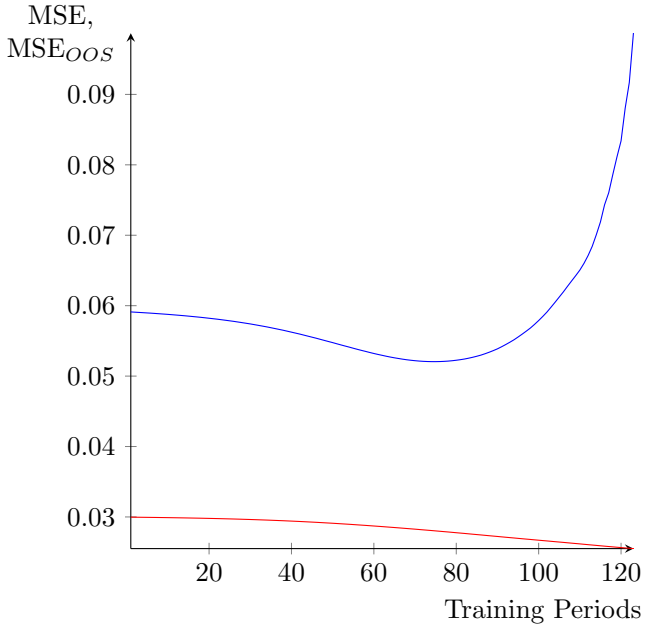


Figure 3.5: ANFIS overtraining, MSE (red) and MSE$_{OOS}$ (blue).

When it comes to the training of the ANFIS it is crucial to consider the amount of iterations of the learning method, the so called training periods. The effect of the training periods on the ANFIS's MSE can be seen in figure 3.5. Similar considerations as to the phenomenon of overfitting described in section 3.4.1 has to be taken into account here. The figure shows that the MSE decreases with each additional training

period. This reflects the fact that the ANFIS's fitting of the observations increases. The MSE_{OOS} however only decreases to the 75th training period and increases afterwards. This reflects the fact that the ANFIS begins to rather fit only the observations than the underlying relationship to predict. This phenomenon in the context of training periods is called overtraining. Therefore the ANFIS will be trained for 75 training periods in this thesis.

3.4.4 ANFIS Results

As explained in section 3.4.2 the ANFIS used is limited to only two input variables. The data available contains six possible input variables though. To fully utilize the available data all combinations of two-pair input variables out of the six possible input variables – in total 15 combinations – will be used to build different ANFIS models.

3.4.4.1 1-year Period

Variable 1	Variable 2	R^2_V	Replaced by Hist. Average
i_{10}	P/E	-0.0612	12
r_{lag}	i_{10}	-0.0719	5
r_{lag}	P/E	-0.1724	3
i_1	P/E_6	-0.1834	4
i_{10}	P/E_6	-0.2435	12

Table 3.4: Five best performing models with input pairs for the 1-year period.

The results of the estimated models over a 1-year period are shown in table 3.4. The table displays the five two-input combinations that have the highest R^2_V. Additionally the table shows the amount of forecasts which had to be replaced by the historical average. The amount of replacements is important to evaluate the R^2_V. The definition of the R^2_V in equation 3.2.1 shows that a model replacing all forecasts by the historical average would score a R^2_V of 0. This would counteract the goal to find an alternative to the historical average. Therefore only a reasonable amount of forecasts should be replaced by the historical average.

The input combination of i_{10} and P/E performs best in forecasting the return. The negative R^2_V indicates that the model is not able to outperform the historical average as predictor though.

Figure 3.6 shows the surface of the estimated model using all 140 observations. Figure 3.7 explains this estimated model in more detail by illustrating different properties from a vertical view. Figure 3.7a shows the white dotted observations which are in the initial training dataset and the purple dotted observations used as input for forecasting.

Figure 3.7b shows the first estimated ANFIS model for the year 1942 using the initial training dataset represented by the white dots. Figure 3.7c visualizes the squared residuals between the forecasted return and the actual return. The size of the squared

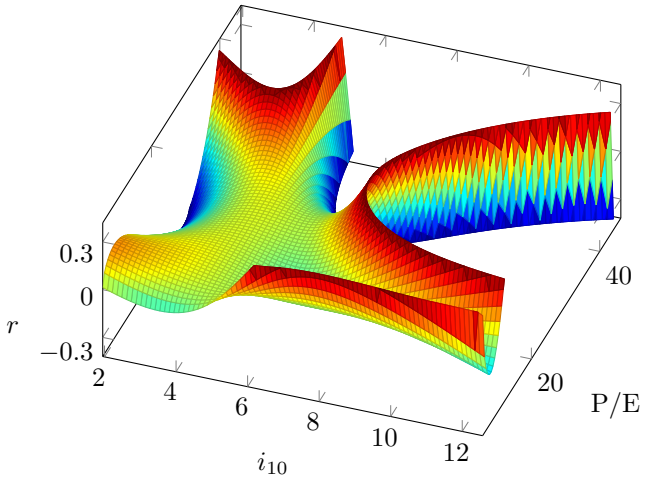
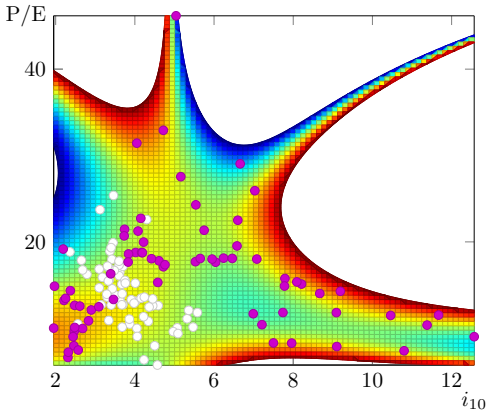
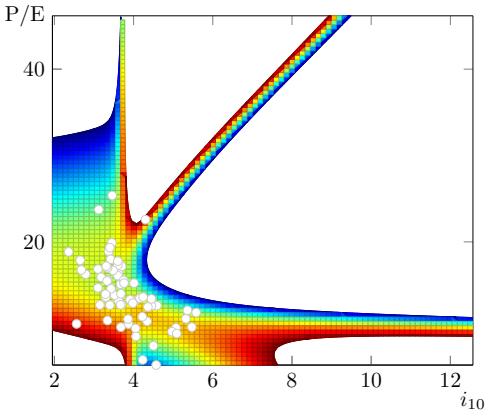


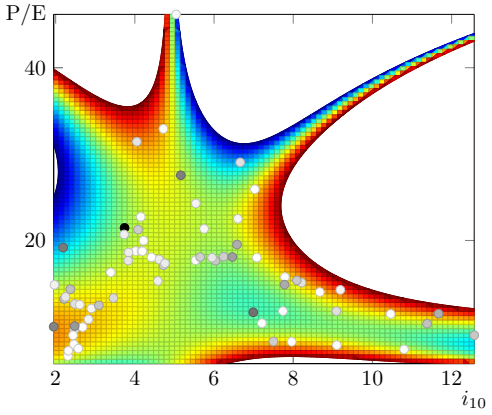
Figure 3.6: Surface of the best performing ANFIS.



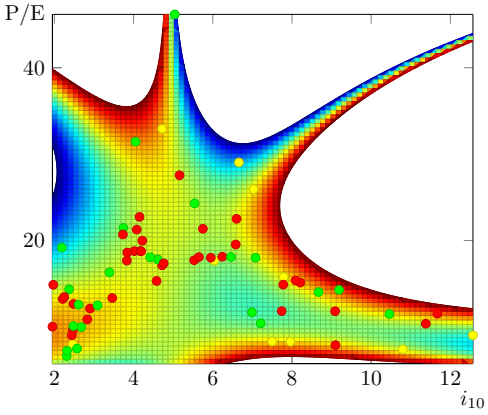
(a) Initial training dataset (white), remaining observations (purple).



(b) Initially trained ANFIS and initial training dataset (white).



(c) Size of squared residuals from small (white) to large (black).



(d) Forecast performance to historical average: worse (red), better (green), replaced by historical average (yellow).

Figure 3.7: Additional information on the trained ANFIS.

residuals can be seen as an indicator in which areas the model performs well and where it performs badly. The observations are here coloured on a scale from white to black. The darker the colour of a dot, the larger is the size of the squared residual. Figure 3.7d shows a direct comparison between forecast and historical average in terms of difference to the actual return. In the case that the historical average is closer to the actual return the dot is marked red. In the case that the forecast is closer to the actual return the dot is marked green. Yellow dots mark the case where the forecast of the ANFIS has been replaced by the historical average as seen in equation 3.4.3.

There are some conclusions to draw from the information of the four sub-figures in figure 3.7. The observations of the initial training dataset are locally clustered resulting in large areas of the estimation replaced by the historical average. By contrast the observations from 1942 onwards are far more scattered. The forecast of the return for these observations is often based on the information of only a few local observations or even replaced by the historical average. The residuals do not show any particular areas where the ANFIS excels. This impression is confirmed by the observation that no areas can be found performing well in comparison to the historical average.

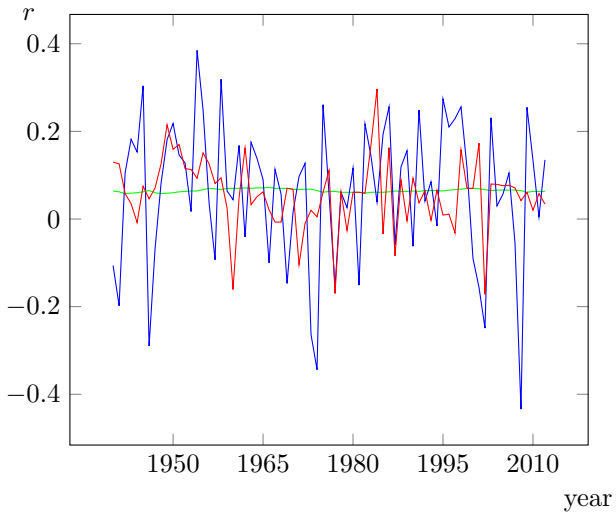


Figure 3.8: Actual return (blue), 1-year ANFIS forecast (red), historical average (green).

Figure 3.8 displays the 1-year actual returns in comparison to the forecasts by the historical average and the ANFIS. While the historical average only changes little the ANFIS forecast is more volatile by comparison. In general the 1-year period ANFIS fails to capture the actual returns though, particularly when large negative returns occur such as in 2007.

3.4.4.2 2-year Period

The following part presents the estimated models over a 2-year period. Table 3.5 shows the results for the five two-input combinations with the highest R_V^2. Three models are

able to forecast better than the historical average. In the following the two best models are presented.

Variable 1	Variable 2	R^2_V	Replaced by Hist. Average
r_{lag}	P/E_6	0.1436	5
r_{lag}	i_{10}	0.1138	8
i_1	P/E_6	0.0238	9
i_{10}	P/E_6	-0.0739	14
i_{10}	P/E	-0.0777	16

Table 3.5: Five best performing models with input pairs for the 2-year period.

Figure 3.9 shows the surface of the model that has the highest R^2_V. It uses the r_{lag} and P/E_6 as input variables. The colour code for the dots in figure 3.10 is identical to the one used for the 1-year model in figure 3.7.

The amount of observations is only half the size of the previous example due to the 2-year periods. Figure 3.10a shows that the white dotted observations of the initial training dataset and the purple dotted observations used to forecast are roughly similar clustered in the same area. In figure 3.10c it can be seen that the model performs very well in terms of the residual size for observations with a P/E_6 less than 15. Figure 3.10d confirms this impression since the historical average is almost always outperformed. The performance of the forecasts for observations with a P/E_6 greater than 15 however is not that clear anymore.

Figure 3.11 displays the 2-year actual returns in comparison to the forecasts by the historical average and the ANFIS. The historical average almost stays unchanged. The ANFIS fails to predict a large negative return in the 1970s. It is able however to capture two large negative returns in the period of the 2000s but one of them only to some extend.

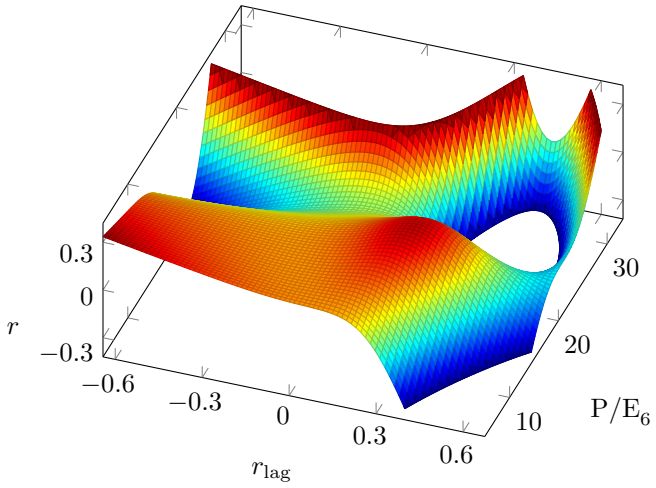
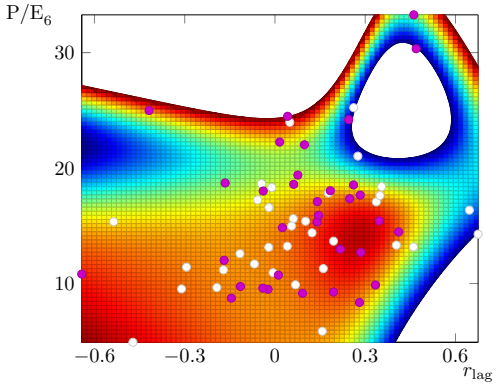
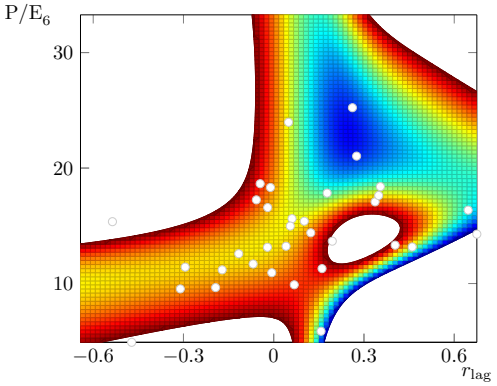


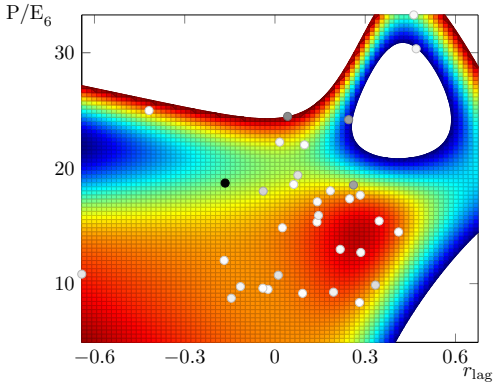
Figure 3.9: Surface of the best performing ANFIS.



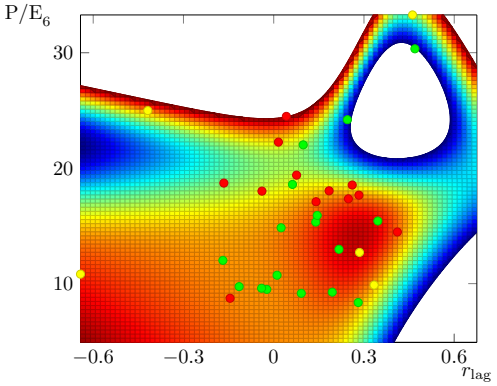
(a) Initial training dataset (white), remaining observations (purple).



(b) Initially trained ANFIS and initial training dataset (white).



(c) Size of squared residuals from small (white) to large (black).



(d) Forecast performance to historical average: worse (red), better (green), replaced by historical average (yellow).

Figure 3.10: Additional information on the trained ANFIS.

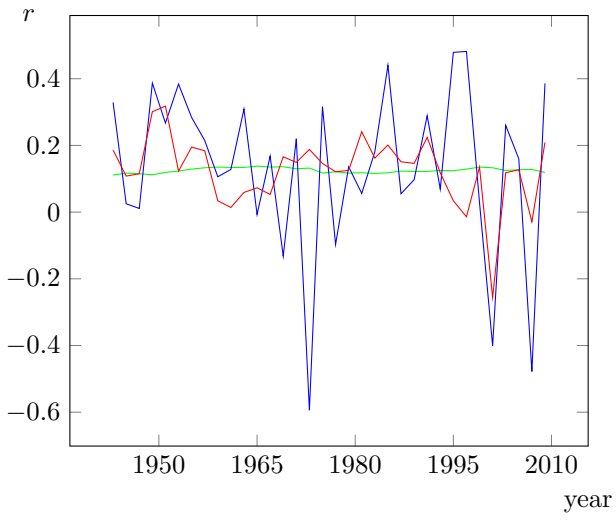


Figure 3.11: Actual return (blue), 2-year best ANFIS forecast (red), historical average (green).

The next part describes the ANFIS over a 2-year period with the second highest R^2_V. It uses the r_{lag} and i_{10} as input variables. Figure 3.12 shows the surface of the estimated model using all 140 observations. Figure 3.13 shows different properties for interpreting the results of the estimation from a vertical view.

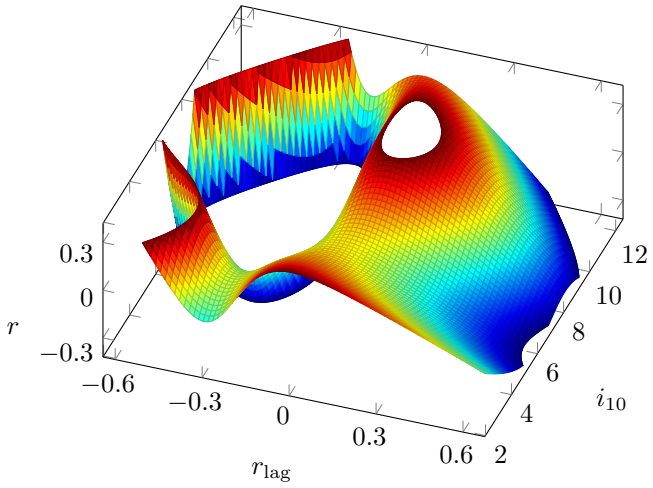
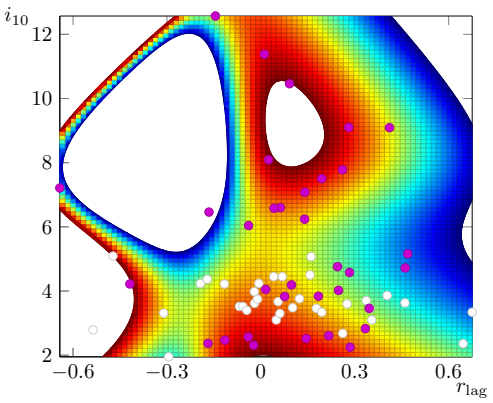


Figure 3.12: Surface of the second best performing ANFIS.

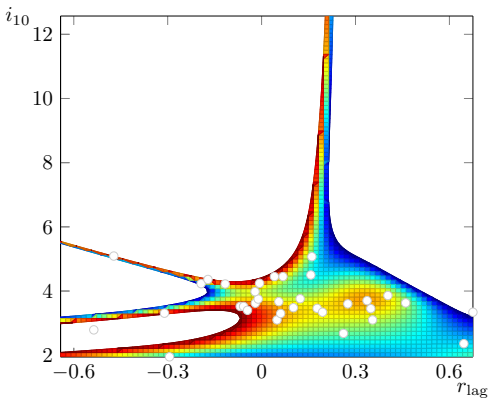
Figure 3.13a illustrates that all of the observations in the training dataset and about half the observations after 1941 are located in the area of values for i_{10} of 6 and less. The other half of the observations after 1941 however has values for i_{10} of 6 and greater. Figure 3.13b illustrates the situation that the estimated model for 1942 almost only covers the area with values for i_{10} of 6 and less. Therefore it is not surprising to see in figure 3.13d that most forecasts for observations with values for i_{10} of 6 and greater are replaced by the historical average.

It is difficult to evaluate the model's performance compared to the historical average for observations with values for i_{10} of 6 and greater. For observations with values for i_{10} of 6 and less the interpretation becomes easier. Although figure 3.13d shows an underperformance of the forecasts compared to the historical average for some observations, figure 3.13c still displays rather small residuals for these forecasts. This leads to the conclusion that the model excels particularly in the area for all observations with a i_{10} of 6 and less.

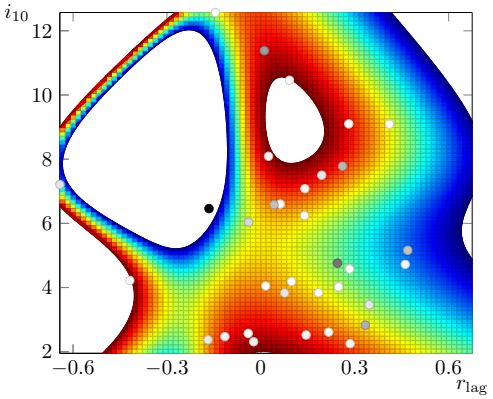
The forecasts of the model, the historical average and the actual returns can be seen in figure 3.14. The period between 1970 and 1985 is characterized by high 10-year US treasury yields. Here it is clear to see that the model replaces the forecasts by the historical average since there is not enough information available for reliable ANFIS forecasts.



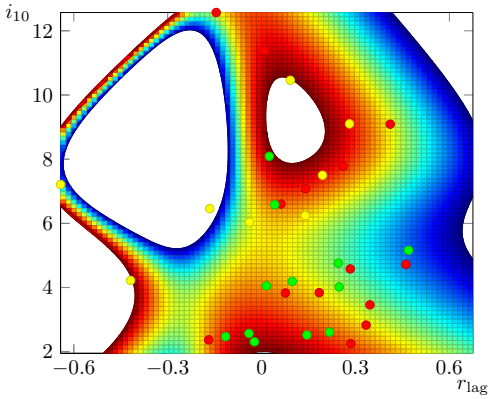
(a) Initial training dataset (white), remaining observations (purple).



(b) Initially trained ANFIS and initial training dataset (white).



(c) Size of squared residuals from small (white) to large (black).



(d) Forecast performance to historical average: worse (red), better (green), replaced by historical average (yellow).

Figure 3.13: Additional information on the trained ANFIS.

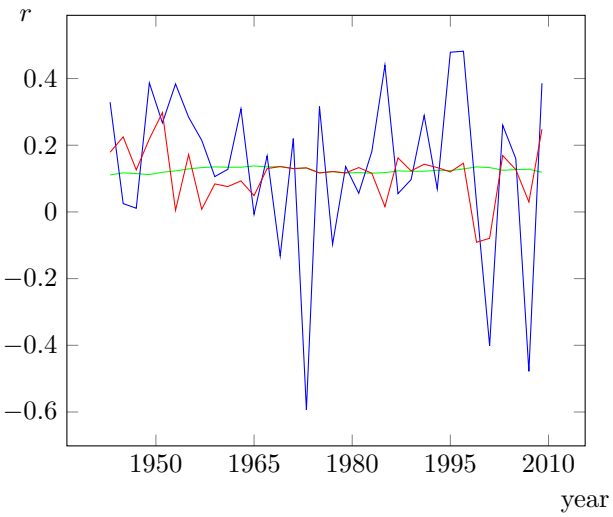


Figure 3.14: Actual return (blue), 2-year second best ANFIS forecast (red), historical average (green).

4 Summary

This thesis provides two ANFIS models clearly able to outperform the historical average as a prediction for S&P500 returns. Therefore the finding of Welch and Goyal (2008) stating the superiority of the historical average as predictor for returns has to be questioned. Nevertheless there are some limitations in the results of this thesis that have to be discussed.

In total 30 different ANFIS models are examined. This large number of models is due to the lack of knowledge of possible nonlinear relationships between explanatory variables and the returns. Therefore various different combinations of explanatory variables are examined to evaluate which combinations have explanatory power.

Of the 30 ANFIS models one half processes data capturing 1-year periods and the other half processes data capturing 2-year periods. The results for the performance of the ANFIS models in predicting returns are mixed. None of the models trained with 1-year period data was able to beat the benchmark of the out-of-sample historical average. Both of the two models able to clearly outperform the out-of-sample historical average are from the 15 ANFIS models trained with data capturing 2-year periods.

The explanation for these results might be found in the used explanatory variables as well as in the captured time horizon.

The findings in this thesis are consistent with the arguments of practitioners such as the value-oriented investors Graham and Dodd (1934). They argue that valuation ratios are an indicator for the prospects of an asset only over a longer time horizon. According to them the prospects over a short time horizon is rather influenced by the sentiment of the market, which includes psychological factors as well. Since valuation ratios do not capture the sentiment of the market they are not able to predict price movements over a short time horizon.

The two best performing 2-year period models both include the lagged return as one of their two explanatory variables. This might be surprising at first sight since the AR(1) model is not able to detect a significant linear influence of the lagged return on the return. In a nonlinear environment however the lagged return seems to play an important role in predicting the return. The best performing 2-year model additionally includes the smoothed P/E ratio as explanatory variable. This is also consistent with Graham and Dodd who state the smoothed P/E ratio as a highly qualified indicator for the long-term performance of an asset.

The second best performing 2-year model additionally includes the 10-year US treasury yield. The findings in this thesis are consistent with Campbell (1987) who stated an influence of long-term US treasury yields on the returns. Nevertheless the found relationship in this thesis shows only a nonlinear relationship in combination with the lagged returns.

When assessing the ANNs, in particular the ANFIS, as a suitable model for financial applications first a look on ANNs in general in necessary. ANNs are high parametric models able to model nonlinear relationships. A main problem of ANNs lies in the difficulties of identifying their parameters. There is no closed form solution available for this task. Therefore iterative approaches searching the parameter space for optimal solutions such as the gradient descent are necessary. Iterative approaches have two inherent weaknesses though. First, an increasing parameter space causes a larger space to be searched resulting in increasing computational expenses. Second, the methods can not distinguish between a local or a global minimum. Therefore the minimum found might only be local.

A partial solution for the difficulties in identifying parameters in the ANN is the learning method HLR presented in this thesis. The first weakness of iterative approaches can be reduced by identifying some of the ANN's parameters by LSE. Therefore the parameter space to be searched by gradient descent is reduced. The second weakness however remains, since some parameters still have to be identified iteratively resulting in possibly only finding a local minimum.

Nevertheless the HLR reduces the computational time for training significantly as stated by Jang (1993). The HLR is applicable to all ANNs linear in some parameters. To apply the HLR however the used software needs to be instructed which of the parameters are linear. This is easily determined in an ANFIS due to the characteristic ANFIS structure as explained in section 2.3.1. In general however ANNs do not have a characteristic structure. Therefore linear parameters would have to be stated individually by the software user which would be a laborious but theoretically possible task. Nevertheless an implemented software solution for the HLR exists so far only for the ANFIS which is one of the motivations to choose the ANFIS in this thesis.

In general the question arises whether high parametric models such as ANFIS are suitable to model 1-year or 2-year returns nonlinearly. The amount of parameters in the ANFIS grows exponentially to the power of the amount of used explanatory variables. A high parametric model might cause an overfitting though, depending on the amount of observations in a dataset. Therefore the ANFIS in this thesis is restricted to only two explanatory variables due to the limitation of only 143 yearly observations. In general the need for large datasets increases with the amount of used parameters in a model. For modelling 1-year or 2-year returns the amount of observations available is limited though, which restricts the use of high parametric models such as the ANFIS as well.

To use the ANFIS with an increased amount of explanatory variables it is crucial to analyze larger datasets. A possible field of application is high frequency trading. In high frequency trading vast amounts of observations are available. Price movements in this area are often driven by short-term factors. Therefore other explanatory variables than the long-term oriented variables used in this thesis should be considered. Short-term oriented variables such as short-term volatility or trading volume might be a fruitful area for further research using the ANFIS.

Bibliography

Ang, Andrew, Piazzesi, Monika, and Wei, Min (2006). "What does the yield curve tell us about GDP growth?" In: *Journal of Econometrics* 131.1, pp. 359–403.

Aström, Karl J and Wittenmark, Bjorn (2011). *Computer-controlled systems: theory and design*. Courier Corporation.

Boole, George (1854). *An Investigation of the Laws of Thought*.

Butler, Alexander W., Grullon, Gustavo, and Weston, James P. (2005). "Can managers forecast aggregate market returns?" In: *The Journal of Finance* 60.2, pp. 963–986.

Campbell, John Y (1987). "Stock returns and the term structure". In: *Journal of financial economics* 18.2, pp. 373–399.

Campbell, John Y. and Shiller, Robert J. (1988). "Stock prices, earnings, and expected dividends". In: *The Journal of Finance* 43.3, pp. 661–676.

Campbell, John Y. and Shiller, Robert J. (1998). "Valuation ratios and the long-run stock market outlook". In: *The Journal of Portfolio Management* 24.2, pp. 11–26.

Campbell, John Y. and Thompson, Samuel B. (2008). "Predicting excess stock returns out of sample: Can anything beat the historical average?" In: *Review of Financial Studies* 21.4, pp. 1509–1531.

Chen, Qingqing and Hong, Yongmiao (2010). "Predictability of equity returns over different time horizons: a nonparametric approach". In: *Working Paper, Cornell University*.

Cochrane, John H. (1999). *New facts in finance*. Tech. rep. National bureau of economic research.

Fama, Eugene F. (1965). "Random Walks in Stock Market Prices". In: *Financial Analysts Journal*, pp. 55–59.

Fama, Eugene F. (1970). "Efficient capital markets: A review of theory and empirical work". In: *The journal of Finance* 25.2, pp. 383–417.

Fama, Eugene F. and French, Kenneth R. (1988a). "Dividend yields and expected stock returns". In: *Journal of financial economics* 22.1, pp. 3–25.

Fama, Eugene F. and French, Kenneth R. (1988b). "Permanent and temporary components of stock prices". In: *The Journal of Political Economy*, pp. 246–273.

Goyal, Amit and Welch, Ivo (2003). "Predicting the equity premium with dividend ratios". In: *Management Science* 49.5, pp. 639–654.

Graham, Benjamin and Dodd, David L. (1934). *Security Analysis: Principles and Technique*. 1st Edition. New York and London: McGraw-Hill Book Company, Inc.

Jang, J.-SR. (1993). "ANFIS: adaptive-network-based fuzzy inference system". In: *Systems, Man and Cybernetics, IEEE Transactions on* 23.3, pp. 665–685.

Kendall, Maurice George and Hill, A. Bradford (1953). "The analysis of economic time-series-part i: Prices". In: *Journal of the Royal Statistical Society. Series A (General)* 116.1, pp. 11–34.

Kothari, Smitu P. and Shanken, Jay (1997). "Book-to-market, dividend yield, and expected market returns: A time-series analysis". In: *Journal of Financial Economics* 44.2, pp. 169–203.

Lamont, Owen (1998). "Earnings and expected returns". In: *The Journal of Finance* 53.5, pp. 1563–1587.

Larsen, P. Martin (1980). "Industrial applications of fuzzy logic control". In: *International Journal of Man-Machine Studies* 12.1, pp. 3–10.

Ljung, Lennart (1998). *System identification*. Springer.

Łukasiewicz, Jan (1920). "O logice trójwartościowej". In: *Ruch filozoficzny*. English translation: L. Borkowski, ed. (1965). *On three-valued logic*. Amsterdam: North-Holland, pp. 87–8.

Łukasiewicz, Jan and Tarski, Alfred (1930). "Untersuchungen über den Aussagenkalkül". In: *Comptes Rendus des sances de la Société des Sciences et des Lettres de Varsovie*, pp. 30–50. English translation: J. H. Woodger, ed. (1956). *Logic, Semantics, Metamathematics: Papers from 1923 to 1938 by Alfred Tarski*. Hackett Publishing Company. Chap. Investigations into the Sentential Calculus, pp. 39–59.

Mamdani, Ebrahim H. and Assilian, Sedrak (1975). "An experiment in linguistic synthesis with a fuzzy logic controller". In: *International journal of man-machine studies* 7.1, pp. 1–13.

McCulloch, Warren S. and Pitts, Walter (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Minsky, Marvin and Papert, Seymour (1969). "Perceptron: an introduction to computational geometry". In: *The MIT Press, Cambridge, expanded edition* 19, p. 88.

Nielsen, J.P. and Sperlich, S. (2003). "Prediction of Stock Returns: A new way to look at it". In: *Astin Bulletin* 33.

Pontiff, Jeffrey and Schall, Lawrence D. (1998). "Book-to-market ratios as predictors of market returns". In: *Journal of Financial Economics* 49.2, pp. 141–160.

Rosenblatt, Frank (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6, p. 386.

Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. (1986). "Learning internal representations by error propagation". In: *Explorations in the microstructure of cognition* 1.

Takagi, Tomohiro and Sugeno, Michio (1985). "Fuzzy identification of systems and its applications to modeling and control". In: *Systems, Man and Cybernetics, IEEE Transactions on* 1, pp. 116–132.

Welch, Ivo and Goyal, Amit (2008). "A comprehensive look at the empirical performance of equity premium prediction". In: *Review of Financial Studies* 21.4, pp. 1455–1508.

Werbos, Paul (1974). "Beyond regression: new tools for prediction and analysis in the behavioral sciences". Ph.D. Thesis. Harvard University.

Zadeh, Lotfi A. (1965). “Fuzzy sets”. In: *Information and control* 8.3, pp. 338–353.

Declaration of Authorship

I hereby certify that the thesis I am submitting is entirely my own original work except where otherwise indicated. I am aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

David Winkel

April 13, 2015